Synop Analyzer 2.2.4
User's Guide

Contents

1 Installation, Tips and Tricks, Customization **1**

1.1 Installation Guide . . . 2
1.1.1 System requirements . . . 2
1.1.2 The standard installation process on MS-Windows . . . 2
1.1.3 Installation problems and trouble shooting . . . 4
1.1.4 The standard installation process on Mac OS, Unix and Linux . . . 5
1.1.5 Activating or updating a license key . . . 6
1.1.6 Increasing the available amount of memory . . . 7

1.2 Accessing Relational Databases . . . 8
1.2.1 The JDBC data access interface . . . 8
1.2.2 Supported database management systems (DBMS) . . . 8
1.2.3 Adding JDBC connectivity for a new DBMS . . . 10
1.2.4 Testing your JDBC connection . . . 12

1.3 Customization and Preferences . . . 14
1.3.1 User specific preferences and settings . . . 14
1.3.2 Customizing the workbench appearance . . . 17

2 Data Import Modules **23**

2.1 The Data Source Specification Panel . . . 24
2.1.1 Supported data formats and data sources . . . 24
2.1.2 The 'Input Data' panel . . . 25
2.1.3 The 'active fields' pop-up dialog . . . 27
2.1.4 The 'Settings' pop-up dialog . . . 32
2.1.5 User specified binnings and discretizations . . . 35
2.1.6 Value groupings and variant elimination . . . 36
2.1.7 Name mappings . . . 39
2.1.8 Taxonomies (hierarchies) . . . 40
2.1.9 Joining with auxiliary tables . . . 41
2.1.10 Computed data fields . . . 43
2.1.11 Transactional and streaming data . . . 45

2.2 The Spreadsheet Import panel . . . 48
2.2.1 Importing a simple, tabular spreadsheet . . . 48
2.2.2 Importing spreadsheets with a complex cell structure . . . 48

2.2.3 Reusing spreadsheet import tasks 50

2.3 The Google Analytics Data Import module 52
2.3.1 Google Analytics 52
2.3.2 Reading data via the Google Analytics Reporting API 52
2.3.3 The panel for specifying a Google Analytics data source 54

2.4 Data Transformations 56
2.4.1 Purpose 56
2.4.2 Aggregating (grouping) data records 56
2.4.3 Splitting a data source in two parts 59

3 Data Analysis and Visualization Modules 61

3.1 The Module 'Statistics and Distributions' 63
3.1.1 Purpose and short description 63
3.1.2 The tabular views 63
3.1.3 The histogram charts view 64
3.1.4 The bottom tool bar 66
3.1.5 Detecting and removing perfect tupels 67

3.2 The Module 'Correlations Analysis' 69
3.2.1 Purpose and short description 69
3.2.2 The tabular correlations view 69
3.2.3 The bottom tool bar 70
3.2.4 The correlations matrix view 71

3.3 The Module 'Bivariate Exploration' 73
3.3.1 Purpose and short description 73
3.3.2 The left hand panel: select fields and value ranges 73
3.3.3 The bivariate matrix 75
3.3.4 The circle plot 77
3.3.5 The bottom tool bar 78
3.3.6 Selecting and exploring matrix cells 79

3.4 The Module 'Pivot Tables' 81
3.4.1 Purpose and short description 81
3.4.2 The left hand panel: select fields and value ranges 82
3.4.3 The bottom tool bar 84
3.4.4 The pivot table panel 88
3.4.5 The chart panel 89

3.5 The Module 'Multivariate Exploration' 90
3.5.1 Purpose and short description 90
3.5.2 Understanding the main panel 90
3.5.3 Working with the range selector buttons 92
3.5.4 Working with detail pop-up dialogs für single fields 93
3.5.5 The bottom toolbar 95

3.5.6 Rearranging and suppressing fields . . . 98
3.5.7 Working with detail structure fields . . . 100
3.5.8 Working with set-valued data fields . . . 102
3.5.9 Creating forecasts and what-if scenarios . . . 104

3.6 The Module 'Split Analysis' . . . 106
3.6.1 Purpose and short description . . . 106
3.6.2 Understanding the main panel . . . 106
3.6.3 Working with the range selector buttons . . . 108
3.6.4 Working with detail pop-up dialogs für single fields . . . 109
3.6.5 The bottom toolbar . . . 111
3.6.6 Rearranging and suppressing fields . . . 113
3.6.7 Working with set-valued data fields . . . 114
3.6.8 Optimizing the control data . . . 115
3.6.9 Automatized series of split analyses . . . 119

3.7 The Time Series Analysis and Forecasting module . . . 121
3.7.1 Purpose and short description . . . 121
3.7.2 Required data properties . . . 121
3.7.3 The summary plot . . . 124
3.7.4 The detail plots . . . 124
3.7.5 The bottom tool bar . . . 125
3.7.6 Saving and exporting settings and results . . . 128

3.8 Detecting Deviations and Inconsistencies . . . 131
3.8.1 Purpose and short description . . . 131
3.8.2 The result view . . . 131
3.8.3 Obtaining correction hints . . . 132
3.8.4 The bottom tool bar . . . 133
3.8.5 Interpretation of deviations: untypical data set or data error? . . . 139

3.9 The Associations Analysis module . . . 142
3.9.1 Purpose and short description . . . 142
3.9.2 Input data formats . . . 142
3.9.3 Definitions and notations . . . 144
3.9.4 Basic parameters for an Associations analysis . . . 145
3.9.5 Pattern content constraints ('item filters') . . . 147
3.9.6 Advanced pattern statistics constraints . . . 149
3.9.7 Result display options . . . 151
3.9.8 Pattern verification and significance assurance . . . 156
3.9.9 Applying association models to new data ('Scoring') . . . 157

3.10 The Sequential Patterns Analysis module . . . 161
3.10.1 Introduction to Sequential Patterns Analysis . . . 161
3.10.2 Input data formats . . . 162
3.10.3 Definitions and notations . . . 163
3.10.4 Basic parameters for an Sequential patterns analysis . . . 165

3.10.5 Pattern content constraints ('item filters') . . . 166
3.10.6 Advanced pattern statistics constraints . . . 167
3.10.7 Result display options . . . 169
3.10.8 Applying sequence models to new data ('Scoring') . . . 171

3.11 The Self-Organizing Maps (SOM) module . . . 174
3.11.1 Purpose and short description . . . 174
3.11.2 Basic parameters for SOM trainings . . . 174
3.11.3 Expert parameters for SOM trainings . . . 176
3.11.4 Interpreting the result visualizations . . . 177
3.11.5 Apply SOM models to new data . . . 183
3.11.6 Creating scoring results . . . 184

3.12 The Regression Analysis panel . . . 188
3.12.1 Purpose and short description . . . 188
3.12.2 Parameters for regression analsis . . . 189
3.12.3 The Regression result panel . . . 190
3.12.4 Applying regression models to new data ('Scoring') . . . 192

4 XML API and Task Automization 196

4.1 The XML Application Programming Interface . . . 197
4.1.1 Command line parameters and the command line processor sacl . . . 197
4.1.2 General structure and a simple example of an XML task . . . 198
4.1.3 Reference description of the <InputData> part . . . 198
4.1.4 Reference description of the analysis task part . . . 205

4.2 The command line processor sacl . . . 219
4.2.1 The command line processor sacl . . . 219
4.2.2 XML analysis task specifications . . . 220
4.2.3 Examples . . . 220

4.3 Task automization and workflows . . . 221

4.4 Defining and Running Reports . . . 222
4.4.1 Concept . . . 222
4.4.2 A sample use case . . . 222
4.4.3 The Visual Report Designer . . . 224
4.4.4 Linking Synop Analyzer analysis results . . . 225
4.4.5 Using Stylesheets . . . 227
4.4.6 Creating HTML or PDF Reports . . . 228

5 Step-by-step Tutorials 230

5.1 Tutorial: Customer Intelligence . . . 231
5.1.1 Business Case . . . 231
5.1.2 Advantages of the Synop Analyzer approach to Customer Intelligence 231
5.1.3 Sample Data used in this Tutorial . . . 232

5.1.4 Step 1: Loading the Data 233
5.1.5 Step 2: Obtaining a First Overview 235
5.1.6 Step 3: Multivariate Interactive Data Exploration 237
5.1.7 Step 4: Customer Intelligence with Multivariate Data Exploration . 239
5.1.8 Step 5: Campaign Plannung and Target Group Selection 241
5.1.9 Step 6: Detailed Look at the Interrelation of two Fields 244
5.1.10 Summary 246

6 Glossary **248**

1

Installation, Tips and Tricks, Customization

In this part of the user's guide we discuss issues such as installation, customization, data base access and general troubleshooting issues.

Installation Guide: Quick installation guide and troubleshooting tips for the case that Synop Analyzer can not be started properly after installation.

Accessing Databases: This chapter describes how Synop Analyzer can directly read data from relational database tables via JDBC. It provides troubleshooting hints if there are problems to establish a JDBC connection, and it describes how a user can connect to an arbitrary database management system with JDBC interface even if that system is not among the DBMS which Synop Analyzer supports per default.

Preferences and Customization: This chapter describes how Synop Analyzer can be customized using the XML preferences file `IA_preferences.xml` and the XML textual resource file `IA_texts.xml`.

1.1 Installation Guide

1.1.1 System requirements

Synop Analyzer is a 100% pure Java software which should run on any 32 or 64 bit system architecture and operating system for which a Java run time environment (JRE) 1.6.0 or higher is available.

For computationally demanding analysis tasks, the software uses highly scalable parallel algorithms. More precisely, Synop Analyzer starts several parallel threads which operate on a common memory. Therefore, it profits from multi-CPU servers and from multi-core CPUs.

The software has been tested on

- Microsoft Windows XP, Windows Vista, Windows Server 2003 and 2007 and Windows 7 (32 and 64 bit).
- Mac OS X
- Linux

As an in-memory data analytics software, Synop Analyzer requires a sufficient amount of RAM when working on large data. As a minimum requirement, the Java virtual machine (VM) should allow Java programs to use at least 256 MB of Heap memory, which can be obtained on any computer with at least 1 GB of RAM. For working more comfortably, the Java VM should make accessible at least 1GB of Heap memory, which is feasible on machines with at least 2GB of RAM.

As a general rule of thumb, Synop Analyzer should have access to Java VM heap memory of at least 30% of the size of the largest single table which is to opened with the software. Therefore, if you want to analyze large tables with sizes of 10 GB to 20 GB without sampling, you should work on a 64 bit operating system and with at least 8GB of RAM.

1.1.2 The standard installation process on MS-Windows

The Synop Analyzer installation package on MS Windows consists of an installer executable called `SynopAnalyzer_setup_Windows.exe` and optionally a separate license key file `IA_license_key_***.txt`.

For installing the software start the setup program. The installer displays a couple of dialogs in which you choose the desired installation directory, for example the directory `C:\IA`, and the program and documentation modules to be installed. Then you finish the installation by clicking the buttons **continue** and **finish**.

After this step, the operating system should display a new program group named **Synop Analyzer** when you click **Start** → **Programs**. The Synop Analyzer root directory, for

example c:\IA, should now contain (among others) the following files and subdirectories:

- the readme file README.txt containing release information and last minute bug reports and workarounds,
- the XML resource and preference files IA_texts.xml and IA_preferences.xml,
- two icon files ending with ..._icon32x32.gif and ..._icon64x64.gif,
- two license files for the open source third party libraries JFreeChart, Jackcess, jTDS and Apache POI which are packaged into Synop Analyzer: license-LGPL.txt and license-Apache_2.0.txt,
- the license file license_test.txt containing a warranty disclaimer for the 'free trial version operating mode' of Synop Analyzer without a valid license key
- the license key file IA_license_key_....txt,
- the subdirectory doc containing the documentation and sample data. The directory should contain at least one language-specific subdirectory such as doc/de_DE and a directory doc/sample_data,
- the subdirectory JDBCTest containing a test package for testing database connections via JDBC,
- if your install package comprises the graphical workbench, the executable file SynopAnalyzer.bat, the debug version SynopAnalyzer_debug.bat and the Java library IA.jar,
- if your install package comprises the command line processor, the executable file sacl.bat and the Java library iacl.jar.

You can access and read the documentation in directory doc without starting the Synop Analyzer software. Just click on the file index.html in your preferred language, for example c:\IA\doc\en_US\index.html in order to open it in your web browser.

The subdirectory doc/sample_data contains several sample data sets which can be used for the first steps when exploring the software features. These data sets are also used in many application examples discussed in the program documentation, in particular in the tutorials.

Once you have created and filled your new Synop Analyzer root directory, the software should be ready to work with.

- SynopAnalyzer.bat starts the Synop Analyzer workbench (GUI).
- sacl.bat starts the Synop Analyzer command line processor.

If you are running the software on a computer with more than 2 GB of RAM and if you want to explore large data files or tables with sizes of several GB or more, you should increase the maximum amount of heap memory which is accessible for `SynopAnalyzer.bat`. To that purpose, edit the batch file and replace the parameter `-Xmx1024m` - which limits the available heap memory to 1 GB = 1024 MB - to a larger value, for example 50% to 75% of the server's total installed RAM. If you want to raise the limit to 8 GB, the content of `SynopAnalyzer.bat` should look like this:

```
java -Xms256m -Xmx8192m -jar IA.jar %1 %2
```

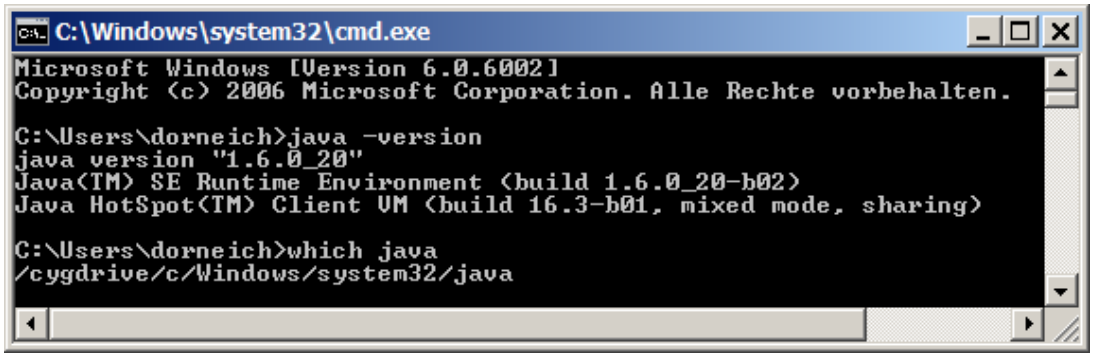
After increasing the Xmx value, you should once try to start the debug version of Synop Analyzer, `SynopAnalyzer_debug.bat` in order to find out whether the system accepts the increased heap limit. If you get an error message, you might have to reduce the upper heap limit. If you get an error message even though the limit is far less than the computer's installed RAM, contact your system administrator: possibly, some restrictive settings of the Java virtual machine prohibit the allocation of more RAM.

If you want to enable Synop Analyzer to read data directly from your relational database management system (DBMS) - for example Oracle, IBM DB2, Teradata, MySQL, etc. - you might have to copy a suitable JDBC driver library for your DBMS - for example `ojdbc6.jar` for Oracle - into your Synop Analyzer installation directory. See `Accessing Relational Databases` for more details.

1.1.3 Installation problems and trouble shooting

You might have performed the standard installation steps as described in section The standard installation process on MS-Windows, but the Synop Analyzer GUI does not come up when double-clicking on `SynopAnalyzer.bat`.

In this case, you should first check whether a suitable Java runtime environment (JRE) is available on your machine. Open a MS-Windows command line box ('Start'–>'Execute'—>'cmd') and issue the following command: `java -version`. The result should look like the following, showing a Java version of 1.6.0 or higher.

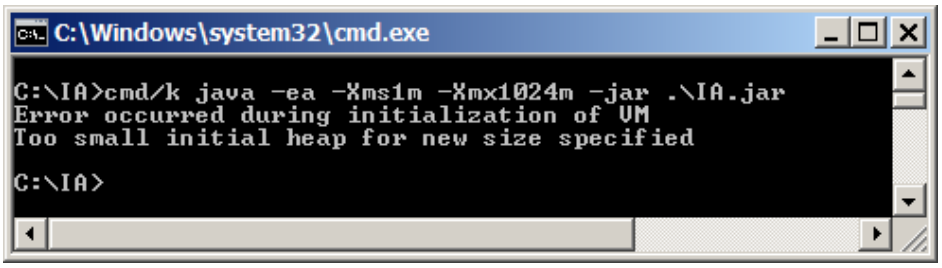

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\dorneich>java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Client VM (build 16.3-b01, mixed mode, sharing)
C:\Users\dorneich>which java
/cygdrive/c/Windows/system32/java

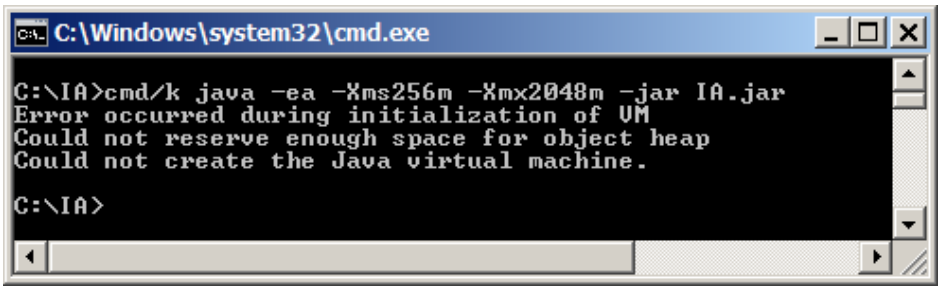
If you know that a more cecent Java version than the one shown in the command box exists on your machine, you can type `which java` in order to see the installation path from which the current default Java version is loaded. (see picture above).

If you can't replace the older default Java installation, you can write the fully qualified directory path to a newer Java version into the two batch files `SynopAnalyzer.bat` and `sacl.bat`, for example

```
c:\Progra~1\java\jre6\bin\java -Xms256m -Xmx1024m -jar IA.jar %1 %2
```

If your Java version is ok but `SynopAnalyzer.bat` does still not start properly, you can invoke `SynopAnalyzer_debug.bat` instead. That version performs some additional checks, and shows its error messages in a black MS-Windows command line box which remains open after the termination of the program call. The error message might involve either the minimum or the maximum heap memory limit


C:\Windows\system32\cmd.exe
C:\IA>cmd/k java -ea -Xms1m -Xmx1024m -jar .\IA.jar
Error occurred during initialization of VM
Too small initial heap for new size specified
C:\IA>


C:\Windows\system32\cmd.exe
C:\IA>cmd/k java -ea -Xms256m -Xmx2048m -jar IA.jar
Error occurred during initialization of VM
Could not reserve enough space for object heap
Could not create the Java virtual machine.
C:\IA>

In both cases, edit the batch file `SynopAnalyzer_debug.bat` and increase the parameter `-Xms` (in the first case) or reduce the parameter `-Xmx` (in the second case) until the error message disappears. Afterwards, repeat the same change in the batch files `SynopAnalyzer.bat` and `sacl.bat`.

1.1.4 The standard installation process on Mac OS, Unix and Linux

The Synop Analyzer installation package for Mac OS, UNIX and Linux consists of an archive file `SynopAnalyzer_setup_MacLinux.zip</but> and optionally a separate license key file <file>IA_license_key_***.txt`. You have to unzip the archive file to an installation directory of your choice. Then, copy the license file into that directory.

After unpacking the zip archive, the installation directory contains two executable shell scripts `SynopAnalyzer.sh` and `sacl.sh` with which you can start the graphical workbench respectively the command line processor of Synop Analyzer.

The other files and directories which have been created in the installation directory are identical to the MS Windows installation; they have been desribed here.

1.1.5 Activating or updating a license key

If you started working with Synop Analyzer by downloading the free trial version then you are working without a license key, and the software will become unusable at the end of the second month after the download.

You can check your current license status by clicking on the **Help** → **About** button in the main menu of the Synop Analyzer GUI.

30 days before your current license expires, Synop Analyzer starts showing the warning message 'your license will expire in xx days' in the title bar of the GUI window.

Once you have decided to acquire a new academic or commercial license, or a temporal extension of your current license, you will be sent a license key file which contains information on

- the license type (non-commercial or commercial; per-user or per-CPU or unlimited),
- the license holder (company or person name),
- the software product name and the vendor,
- the modules and software functions activated by the license,
- the maximum number of installations (CPUs or users) covered by the license,
- the license expire date.

The file name of the license file contains an abbreviation of the license holder name and the license expire date, for example `SynopAnalyzer_license_key_SampleInc_Dec2010.txt` if the license holder is Sample Inc. and the license expires on December 31, 2010. Do not modify the content of the license file in any kind, otherwise the license key will become unusable.

As long as the expire date of your current license has not been reached, you can activate a new license key file very easily via the main menu path **Preferences** → **General Preferences** → **license file**. After assigning the new license file, please restart Synop Analyzer and check the **Help** → **About** menu item in order to verify your new license features.

Once your current license has expired, you have to activate the new license key file by manually editing your Synop Analyzer preferences file. The preferences file resides in the root directory of your Synop Analyzer installation and is named (unless you have renamed it) `IA_preferences.xml`. Search the settings parameter

```
<Setting name="licenseFile" ...  />
```

and modify its `value` attribute so that it contains the new license file:

```
<Setting name="licenseFile" ...
 value="IA_license_key_SampleInc_Dec2013.txt" ...  />
```

If you store the license file in another directory than the Synop Analyzer root directory, `value` must contain the fully qualified path name to the new license file.

1.1.6 Increasing the available amount of memory

If you are running the software on a computer with more than 2 GB of RAM and if you want to explore large data files or tables with sizes of several GB or more, you should increase the maximum amount of heap memory which is accessible for `SynopAnalyzer.bat`. To that purpose, edit the batch file and replace the parameter `-Xmx1024m` - which limits the available heap memory to 1 GB = 1024 MB - to a larger value, for example 50% to 75% of the server's total installed RAM. If you want to raise the limit to 8 GB, the content of `SynopAnalyzer.bat` should look like this:

```
java -Xms256m -Xmx8192m -jar IA.jar %1 %2
```

After increasing the Xmx value, you should once try to start the debug version of Synop Analyzer, `SynopAnalyzer_debug.bat` in order to find out whether the system accepts the increased heap limit. If you get an error message, you might have to reduce the upper heap limit. If you get an error message even though the limit is far less than the computer's installed RAM, contact your system administrator: possibly, some restrictive settings of the Java virtual machine prohibit the allocation of more RAM.

1.2 Accessing Relational Databases

1.2.1 The JDBC data access interface

The JDBC application programming interface is the industry standard for database-independent connectivity between Java programs and a wide range of databases. Synop Analyzer uses this standard for reading data directly from database tables. Each database management system (DBMS) requires a specific JDBC driver in the form of a Java library (jar-file) for providing JDBC connectivity.

For a couple of widely used DBMS, a suitable JDBC driver comes with the Synop Analyzer install package. These java libraries are not part of Synop Analyzer and not covered by your Synop Analyzer license and support agreement. They are free software which has been placed into the public domain by their authors under theGNU Lesser Public License (GLPL).

The license conditions of other widely used DBMS do only permit the distribution of JDBC drivers together with a license of the underlying DBMS. For these databases, Synop Analyzer does not install the JDBC driver but relies on a preexisting JDBC driver installation on the database server. Nontheless, Synop Analyzer is preconfigured for using these JDBC drivers. Both groups of 'known' DBMS are described in section Supported DBMS

If you are working with a DBMS which is not part of the list of 'known' DBMS, you can manually configure Synop Analyzer for reading data from this new DBMS by editing the preferences file `IA_preferences.xml`. A step-by-step instruction for declaring a new DBMS can be found in section Adding a new supported DBMS.

If you want to test whether a given table in a given database of a given DBMS can be accessed from Synop Analyzer using given user and password credentials, you can use a separate JDBC connection tester program called `JDBCTest.bat` which comes with Synop Analyzer. The usage of this program is described in section Testing your JDBC connection.

1.2.2 Supported database management systems (DBMS)

- **Microsoft Access**

Driver library:	`jackcess-1.2.1.jar` (included in the Synop Analyzer install package)
Download URL:	http://jackcess.sourceforge.net/
License:	LGPL (GNU Lesser Public License)
Install instructions:	Nothing to do.

- **Microsoft SQL Server**

Driver library:	`jtds-1.2.4.jar` (included in the Synop Analyzer install package)
Download URL:	http://jtds.sourceforge.net/
License:	LGPL (GNU Lesser Public License)
Install instructions:	Nothing to do.

- **Oracle**

Driver library:	`ojdbc6.jar`
Download URL:	http://www.oracle.com/technetwork/database/features/-jdbc/index-091264.html
License:	OTN (Oracle Technology Network License)
Install instructions:	Find the driver library on your database server or download it. Copy the driver library into the Synop Analyzer install directory.

- **IBM DB2**

Driver library:	`db2jdbc4.jar`
Download URL:	http://www-01.ibm.com/software/data/db2/express/-download.html
License:	IBM-specific no-charge license
Install instructions:	Find the driver library on your database server or download it. Copy the driver library into the Synop Analyzer install directory.

- **Teradata**

Driver libraries:	`tdgssconfig.jar, terajdbc4.jar`
Download URL:	http://www.teradata.com/downloadcenter/
License:	Teradata license
Install instructions:	Find the driver libraries on your database server or download them. Copy the driver libraries into the Synop Analyzer install directory.

- **Sybase**

Driver library:	`jtds-1.2.4.jar` (included in the Synop Analyzer install package)
Download URL:	http://jtds.sourceforge.net/
License:	LGPL (GNU Lesser Public License)
Install instructions:	Nothing to do.

- **Progress**

Driver libraries:	`base.jar openedge.jar pool.jar spy.jar util.jar`
Download URL:	No free download available. The libraries come with the database Progress 10.x
License:	See your Progress 10.x license
Install instructions:	Find the driver libraries on your database server. Copy the driver libraries into the Synop Analyzer install directory.

- **MySQL**

Driver library:	`mysql-connector-java-5.x.x-bin.jar`
Download URL:	http://dev.mysql.com/downloads/connector/j/
License:	GPL (GNU Public License)
Install instructions:	Find the driver library on your database server or download it. Copy the driver library into the Synop Analyzer install directory and rename it to `mysql-connector-java-bin.jar`.

- **PostgreSQL**

Driver library:	`postgresql-9.x-xxx.jdbc4.jar`
Download URL:	http://jdbc.postgresql.org/download.html
License:	BSD (Berkeley Software Development License)
Install instructions:	Find the driver library on your database server or download it. Copy the driver libraries into the Synop Analyzer install directory and rename it to `postgresql.jdbc4.jar`.

- **InterSystems Caché**

Driver library:	`CacheDB.jar`
Download URL:	http://www.intersystems.de/cache/downloads/index.html
License:	Evaluierungs- und Testlizenz
Install instructions:	Find the Java 1.6 version of driver library on your database server or download it. Copy the driver library into the Synop Analyzer install directory.

1.2.3 Adding JDBC connectivity for a new DBMS

If your database management system (DBMS) provides a JDBC interface and driver library but does not figure in the list of 'known' DBMS, you can manually add your DBMS' JDBC driver to the list of supported JDBC connections.

For declaring a new DBMS-JDBC driver combination, you need to have the following information at hand:

1. The name under which the new data source will appear in the list of all available data sources
2. The name of the Java class which implements the JDBC driver (such as `oracle.jdbc.driver.OracleDriver` for Oracle)
3. The first part of the JDBC connection string which precedes the host name (such as `jdbc:oracle:thin:` for Oracle)
4. The hostname prefix within the JDBC connection string. This is `//` for most JDBC drivers and `@` for Oracle.

5. Each JDBC driver has a different default port number via which it communicates with the database. If your database user must use another port, you must know that port number.

6. The middle part of the JDBC connection string which follows the host name or port number and precedes the database name (for example : for Oracle, / for many other JDBC drivers or `;databaseName=` for Progress/Openedge)

7. The SQL statement for detecting the column names and types in a given table. The statement must return the column names in the first column and the table types in the second column of its result set.
   Examples: `SELECT COLUMN_NAME, DATA_TYPE FROM ALL_TAB_COLS WHERE TABLE_NAME='<tablename>' AND OWNER='<schema>'` for Oracle or `SHOW COLUMNS FROM <schema>.<tablename>` for MySQL.

8. The SQL statement for detecting the occupied disk space of table in bytes.
   Examples: `SELECT AVG_ROW_LEN*NUM_ROWS "data_length" FROM DBA_TABLES WHERE TABLE_NAME='<tablename>' AND OWNER='<schema>'` for Oracle or `SHOW TABLE STATUS LIKE '<schema>.<tablename>'` for MySQL.

9. The column name containing the table size information within the result set of the above statement for detecting the table size. Example: `data_length`.n

Once you have collected this information from the JDBC driver documentation of your JDBC driver, you can test the correctness of the settings by running a little test program called `JDBCTest.bat` which can be found in the subdirectory `JDBCTest` of the Synop Analyzer installation directory. The usage of this file and its accompanying parameter file `JDBCTest_params.txt` is described in more detail in section Testing your JDBC connection.

When you are sure your settings describe a working JDBC connection to a DMBS which does not figure on Synop Analyzer's list of automatically supported DBMS, you can declare this user-defined JDBC data source to Synop Analyzer. Open the preferences file `IA_preferences.xml` in an arbitrary XML text editor (for example in the freely available general purpose editor Notepad++) and search the following setting:

```
<Setting name="userDefinedDBMSName"
```

This setting is the first out of a series of 9 settings which all start with the prefix `"userDefined`, the last setting within the series is

```
<Setting name="userDefinedTableSizeColumnName".
```

Copy the 9 pieces of information which you have collected in the list shown above between the double quotes following the 9 `value=` attributes of the 9 settings parameters, then save the modified file `IA_preferences.xml`. When you start Synop Analyzer the next time, you will find your newly defined JDBC data access in the pop-up list of available DBMS in the 'database connect' panel.

1.2.4 Testing your JDBC connection

If you encounter problems when accessing data residing in a DBMS (database management system), or if you want to define and test a new JDBC data source, you can test your JDBC connection using the test program `JDBCTest.bat` which resides in subdirectory `JDBCTest` of the Synop Analyzer installation directory.

The test program `JDBCTest.bat` is an executable MS-Windows batch file which can be started by double-clicking on it. The main program comes with a couple of auxiliary and source code files. Legally, the entire JDBCTest package is not a part of Synop Analyzer but has been placed into the public domain under the BSD License, which means that you can do almost anything with it, use it, modify the source code and distribute it - as far as you maintain the original copyright and warranty disclaimer note in the source code and as far as you do not sue the author for any damages which result from using it.

- `JDBCTest.bat` is just a wrapper program which starts the executable Java program `JDBCTest.jar` and keeps the console output window open after the termination of the program so that the output can be read.
- `JDBCTest.jar` is a zipped executable java archive which can be introspected and unzipped by many zip/unzip programs such as `7zip` or `IZArc`.
- `JDBCTest.java` is the Java source code for `JDBCTest.jar`. You only need this file if you have some basic Java programming skills and if you want to extend or modify the program `JDBCTest.jar`.
- `JDBCTest_params.txt` is the parameter file in which you have to adapt a couple of parameters to fit your specific DBMS, JDBC driver, hostname, database name, user name, and password settings.

If you want to test whether your database management system and your JDBC driver is suitable for working with Synop Analyzer, do the following:

1. copy all java libararies of your JDBC driver (for example `ojdbc6.jar` for Oracle or `db2jcc4.jar` for DB2) into the Synop Analyzer installation directory
2. Open the batch file `JDBCTest/JDBCTest.bat` in a text editor, for example Notepad or Notepad++ and make sure all libraries of your JDBC driver appear after the `-cp` option. Use `;` as separator character and don't forget the relative directory path prefix. For example, for adding the Oracle JDBC library you could write `-cp ../ojdbc6.jar`.
3. Edit the parameter file `JDBCTest_params.txt`. Note that all lines starting with '#' are comment lines which will be ignored by the program `JDBCTest.bat`. If you are working with one of the DBMS for which `JDBCTest_params.bat` already contains some (commented-out) settings, activate these settings by removing the '#' and edit them so that your host name, port number, user name, database name and password are correctly inserted. If your DBMS does not occur in `jdbctest_params.txt`, create a new section with settings for your DBMS.

4. Start `JDBCTest.bat` and look at the diagnostic output; refine your parameter settings until the test protocol tells you that your configuration is suitable for Synop Analyzer.n

If you found the test package JDBCTest helpful and if you have added a new DBMS to`JDBCTest_params.txt` or if you have performed a bugfix or an improvement, your feedback to the Synop Analyzer team is appreciated (make sure you anonymize your IP addresses, user names and passwords when you paste in snippets from your parameter file!):

1.3 Customization and Preferences

1.3.1 User specific preferences and settings

This chapter describes how Synop Analyzer can be configured towards the needs of single users by defining user-specific settings in the preferences file `IA_preferences.xml`.

The settings file IA_preferences.xml

Synop Analyzer stores and reads more than a hundred settings, default values and customization parameters in a preferences file named `IA_preferences.xml` residing in the root directory of the Synop Analyzer installation. This file is an XML document conforming to the XML schema`http://www.synop-systems.com/xml/InteractiveAnalyzer-Preferences.xsd`. The structure of the document is quite simple: it consists of a series of tags of the form

```
<Setting name="..." type="..." module="..." value="..."
 default="..."/>,
```

for example

```
<Setting name="smallIcon" type="filename" module="GUI"
 value="IA_icon32x32.gif" default="IA_icon32x32.gif"/>.
```

name is the name of the parameter to be defined, **type** is its data type (int, double, free textual string, file name, or choice list). **module** states the functional modules for which the parameter applies and **value** is the actual value of the parameter.

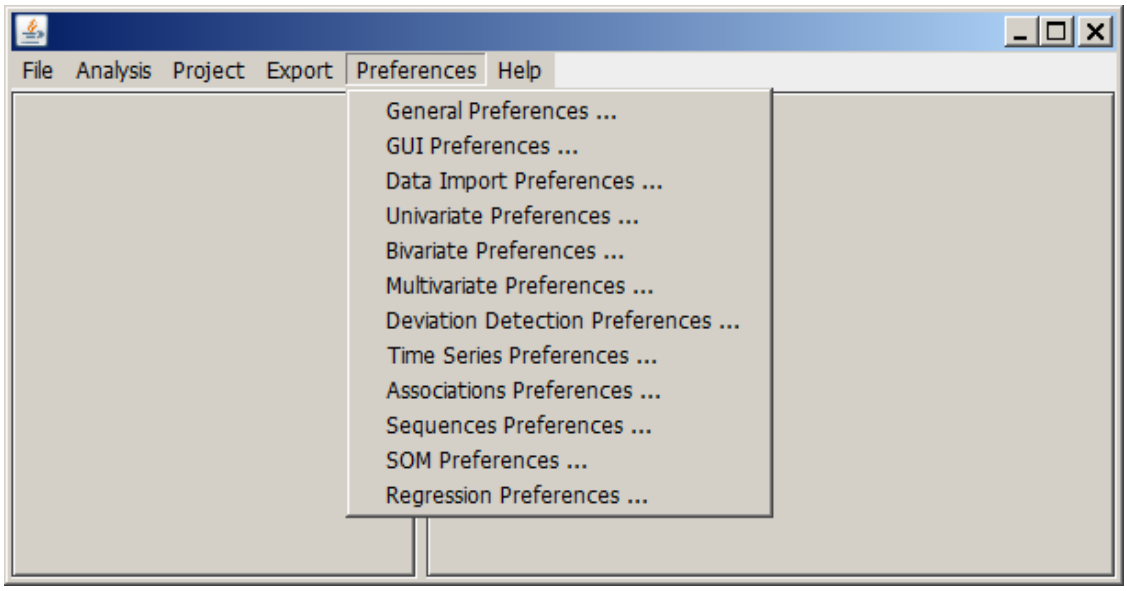
The attribute **default** is ignored when Synop Analyzer parses the preferences file. It serves to memorize the default setting of the parameter at the installation time of the software and helps undoing changes which lead to surprising effects or errors.

If you remove the file `IA_preferences.xml`, Synop Analyzer generates a new version of the file in which all parameter values are identical to the default values.

As a user of the software you can work with your own version of `IA_preferences.xml`. For example, you can copy the original file to your home directory and rename it, e.g. to `c:\users\smith\IA_preferences_smith.xml`. Then you write and save a batch file, e.g. `c:\users\smith\my_IA.bat` which calls Synop Analyzer with the name of the new batch file as (second) command line parameter. (The first command line parameter, which contains the analysis task to be executed automatically on program startup, can remain empty.) The batch file should look like this:

```
c:/IA/SynopAnalyzer.bat "" c:/users/smith/IA_preferences_smith.xml.
```

You can modify the parameters in `IA_preferences.xml` by directly editing the file using an arbitrary text editor. You can also customize `IA_preferences.xml` by means of the menu button **Preferences** in the main menu of the Synop Analyzer graphical workbench.



Default directories

The following parameters in `IA_preferences.xml` specify default directory paths:

- **defaultTempDirectory** defines the directory which will be used by Synop Analyzer as temporary 'RAM disk' while reading and compressing large data files.
- **defaultResultDirectory** defines the default directory for storing analysis results such as generated mining models or analysis task descriptions in XML format.
- **defaultInputFileDirectory** defines the default directory in which flat file input data to be opened in Synop Analyzer are expected to reside.
- **defaultInputSpreadsheetDirectory** defines the default directory in which MS Excel or other OOXML format spreadsheets to be opened in Synop Analyzer reside.

Default settings for connections to relational databases

The following parameters in `IA_preferences.xml` specify default settings for reading data from relational databases:

- **defaultDBMS** specifies the default relational database management system (DBMS). Possible values are `Oracle`, `MySQL`, `Postgres`, `SQLServer`, `Access`, `DB2`, `Sybase`, `Teradata`, `Progress` and `ODBC`. The latter specifies Sun's generic ODBC-JDBC bridge.
- **defaultDBServer** defines the server name of the default database server. Use `localhost` of you primarily work with databases and DBMS installed on your local computer.

- **defaultDBUser**</TT> default database user name to be used for logging in to the database server.
- **defaultDataBase** defines the name of the default database.
- **defaultDBSchema**: here, you can specify the name of a database schema in which most of the tables to be analyzed reside. If you always work with one single table, you can also specify `[schema_name].[table_name]` here.

Debug and trace

The following parameters specify settings for debug and trace. These settings define up to which extent Synop Analyzer produces progress, error and warning information while processing analysis tasks, or while your are interactively working with the graphical workbench. This information can be helpful to track down problems or unexpected program behavior.

- **traceFile**: per default, Synop Analyzer writes progress information to the standard console output ('stdOut'). You can redirect trace output into a file by specifying a file name here.
- **traceLevel** defines the amount of information to be written. Allowed values range from 0 to 3. 0 means that no trace messages are produces as long as no unexpected error occurs or a predefined task could not be executed properly. 3 means that a lot of detailed trace information is produced for each single analysis step, both in the case of an error and in the success case. In trace level 3, the trace file can become very large when running complex tasks on large input data. Use level 3 only temporarily, in order to track down a specific problem, but not permanently in normal operation mode.

Hiding modules

If you or the users for whom you are customizing the software are expected to use only a limited subset of all functional modules offered by Synop Analyzer, you can hide certain modules. This makes the workbench less complex and easier to use since many buttons and expert parameter input fields might disappear from the graphical user interface (GUI).

You can hide modules by setting one or more of the following parameters in `IA_preferences.xml` to "false":

- **activateGUI**: if this parameter is "false", only the command line processor `sacl.bat`, but not the graphical workbench can be used.
- **activateCommandLine**: if this parameter is "false", only the graphical workbench `SynopAnalyzer.bat`, but not the command line processor (and as a consequence no automated task processing in batch mode) can be used.

- **activateDatabaseAccess**: if this parameter is "false", data import from relational databases is impossible.
- **activateSpreadsheetImport**: if this parameter is "false", data import from MS Excel is impossible, and the import wizard for transforming data from spreadsheets with a complex structure into a flat tabular form is not available.
- **activateDataPreparation**: if this parameter is "false", the data preprocessing functions such as filter, collate, sort, compute fields, pivot/unpivot are unavailable.
- **activateUnivariateExploration**: if this parameter is "false", no univariate data exploration, value distribution histograms and data field statistics can be performed.
- **activateBivariateExploration**: switches on/off the bivariate exploration module.
- **activateMultivariateExploration**: switches on/off the multivariate exploration module.
- **activateTestControlVerification**: switches on/off the module for comparing a test and a control data set and for verifying hypotheses concerning differences between the test and the control data.
- **activateCorrelations**: switches on/off the module which calculates correlations between data fields.
- **activateDeviations**: switches on/off the module which finds deviations and presumable data inconsistencies
- **activateTimeSeries**: switches on/off the module for time series analysis and forecast.
- **activateAssociationsTrain**: switches on/off the associations analysis module.
- **activateSequencesTrain**: switches on/off the sequential patterns analysis module.
- **activateSOMTrain**: switches on/off the neural networks module 'SOM' (self organizing maps) for clustering, classification, prediction and deviation detection.
- **activateRegressionTrain**: switches on/off the linear regression module

Note: if some of the modules are deactivated in the initial version of `IA_preferences.xml`, then your license does not enable you to use the modules. In this case, setting the corresponding 'activate...' parameter to "true" has no effect.

1.3.2 Customizing the workbench appearance

This chapter describes how the graphical appearance and the textual labels of the Synop Analyzer workbench can be modified. The description is targeted at

- **End users** who want to personalize the appearance and the look and feel of the software to match their personal preferences.

- **System integrators** who are integrating Synop Analyzer in an existing BI software stack and who want the integrated solution to have a uniform color scheme and look and feel.
- **OEM partners** who are building their own software solutions using Synop Analyzer components.

Technically, the custimizations described here are effectuated by modifying two XML resource files which come with the Synop Analyzer software:

- The preferences file`IA_preferences.xml`,
- The textual resource file `IA_texts.xml` or a renamed and customized substitute for that file which is referenced in the following settings parameter of the preferences file `IA_preferences.xml`:
  `<Setting name="textualResourceFile" type="filename" module="all" value="c:\users\smith\my_personal_IA_texts.xml" default="IA_texts.xml"/>`.

Application name, copyright, license agreement, icons

The file `IA_preferences.xml` contains 5 parameters which control the application name, the application icon, the copyright statement and the short version of the license agreement which is printed at the beginning of the Synop Analyzer trace file.

These parameters, and the possibility to freely access and modify them in the XML file, are targeted at OEM partners who are integrating the Synop Analyzer software into their own software offerings which are sold under the partner's own label, copyright and icon. Note that these entries in the preferences file are matched against the Synop Analyzer license key when the software is started. The software will issue an error message and terminate if the entries found in the preferences file do not match the available license key.

These are the 5 mentioned settings and their default values:

- `<Setting name="application" type="string" module="all" value="Synop Analyzer"/>`
- `<Setting name="copyright" type="string" module="all" value="(C) 2012, 2013 Synop Systems UG (haftungsbeschr?nkt)"/>`
- `<Setting name="licenseAgreement" type="string" module="all" value="Free trial version of Synop Analyzer.\nDisclaimer: The author of this software accepts no responsibility for\ndamages resulting from the use of this product and makes no warranty\nor representation, either express or implied, including but not limited\nto, any implied warranty of merchantability or fitness for a particular\npurpose. This software is provided 'AS IS', and you, its user, assume\nall risks when using it."/>`

- `<Setting name="smallIcon" type="filename" module="GUI" value="IA_-icon32x32.gif"/>`
- `<Setting name="largeIcon" type="filename" module="GUI" value="IA_-icon64x64tr.gif"/>`

Look and feel

Synop Analyzer supports 3 different 'look and feel' (LAF) modes

- the 'Windows' look and feel,
- the 'Java native' or 'Metal' look and feel,
- the 'Motif' look and feel which is familiar to users of X11 GUIs under UNIX/Linux operating systems.

The default setting is 'Windows'. You can activate one of the other 'LAF's by modifying the following entry in your preferences file (`IA_preferences.xml`):

```
<Setting name="lookAndFeel" type="choice" module="GUI" value="windows"
 choices="metal|motif|windows"/>.
```

Color palettes

For all Synop Analyzer workbench panels which show colored charts and other data visualizations, the preferences file `IA_preferences.xml` contains color palette settings which can be modified in order to match the user's or partner's preferences.

For the '**Statistics and Distributions**' panel, the following color parameter is available:

- `<Setting name="barColors" type="string" module="UnivarStats" value="-0:0:255 255:0:0 0:255:0 255:255:0 0:255:255 255:0:255 192:192:192 255:128:0 0:255:128 128:0:255 128:255:0 0:128:255 255:0:128 128:128-:128"/>`.
  Each number triple separated by colons represents one RGB color code with R (red), G (green) and B (blue) values in the range 0 to 255. The first triple specifies the color of the first histogram bar - in the setting shown above, an intense blue - the second triple represents the second histogram bar, and so on.

For the '**Bivariate Exploration**' panel, the following color parameters are available:

- `<Setting name="histogramBarColor1" type="string" module="BivarStats" value="0:0:255"/>`.
  This RGB color code specifies the color of the first, third, fifth, etc. range defined by clicking on the selector bars on the left hand part of the panel.

- `<Setting name="histogramBarColor2" type="string" module="BivarStats" value="255:0:0"/>`.
  This RGB color code specifies the color of the second, fourth, sixth, etc. range defined by clicking on the selector bars on the left hand part of the panel.
- `<Setting name="circleColor" type="string" module="BivarStats" value="0:0:255"/>`.
  This RGB color code specifies the color of the circles in the circle plot on the right hand part of the panel.

For the '**Multivariate Exploration**' and the '**Test/Control Analysis**' panel, the following color parameters are available:

- `<Setting name="histogramBarColorSelected1" type="string" module="MultivarStats" value="0:0:255"/>>`.
  The RGB code of the histogram bars representing the selected data subset respectively the test data subset.
- `<Setting name="histogramBarColorSelected2" type="string" module="MultivarStats" value="255:64:64"/>>`.
  The RGB code of the histogram bars representing the control data subset in the Test/Control Analysis panel.
- `<Setting name="histogramBarColorAll" type="string" module="MultivarStats" value="220:255:220"/>>`.
  The RGB code of the histogram bars representing the entire data (background distribution).
- `<Setting name="selectedTitleColor" type="string" module="MultivarStats" value="0:0:255"/>>`.
  The RGB code specifying the color of the histogram title texts for those histograms in which the user has performed data selections by clicking on the checkbox selector bars below the chart.

Labels, dialog texts and tool tips

All labels, panel titles, message or tool tip pop-up texts appearing in the Synop Analyzer workbench are defined in the textual resource file `IA_texts.xml` or a renamed and customized substitute for that file.

The textual resource file is an XML document conforming to to the XML schema`http://www.synop-systems.com/xml/InteractiveAnalyzerTexts.xsd`.The structure of the document is quite simple: it consists of a series of tags of the form

```
<Label key="..." inGlossary="...">
  <Modules> ...</Modules>
  <Value text="..." lastModified="..." lang="..." target="..."/>
  <Description text="..." lastModified="..." lang="..." target="..."/>
</Label>.
```

The various parts of the `<Label>` tag have the following meaning and functions:

- **key**is the name under which the label is referenced in the program code. You must never change the `key` attribute!
- **inGlossary**specifies whether or not the automatically generated glossary `help-Glossary.html` should contain an entry for the current label.
- **<Modules>**contains a space separated list of the modules or panels in which the label appears.
- **<Value text="..." lang="..." .../>**contains the text which actually represents that label on the panel when the language `lang` is active.
- The optional sub-element **<Description text="..." lang="..." .../>** contains the text which pops up as a tool tip if the mouse pointer is placed on the GUI element carrying the current label. Furthermore, this text appears in the glossary entry created for the label.

The sub-elements `<Value>` and `<Description>` contain an optional attribute **target**. This attribute specifies the application name for which the current label text or label description is being defined. Using this attribute, OEM partners can overwrite any textual resource of Synop Analyzer when embedding Synop Analyzer components into their own applications.

Let us look at an **example**. The **Undo** button in the Test/Control Analysis panel has the following default entry in`IA_texts.xml`:

```
<Label> inGlossary="true" key="Undo">
  <Modules>TestControlSplit</Modules>
  <Value text="Undo" lastModified="2010-03-01" lang="en_US"/>
  <Description text="Undo the previous control data optimization.
    That means, reactivate all available control data records."
    lastModified="2010-03-01" lang="en_US"/>
</Label>
```

The OEM partner offering an application called 'XY Explorer' could modify this label as follows:

```
<Label> inGlossary="false" key="Undo">
  <Modules>TestControlSplit</Modules>
  <Value text="Undo" lastModified="2010-03-01" lang="en_US"/>
  <Value text="Alle" lastModified="2010-04-25" lang="de_DE"
    target="XY Explorer"/>
  <Description text="Undo the previous control data optimization.
    That means, reactivate all available control data records."
    lastModified="2010-03-01" lang="en_US"/>
  <Description text="Wähle alle verfügbaren Kontrolldaten aus."
    lastModified="2010-04-25" lang="de_DE" target="XY Explorer"/>
</Label>
```

The effect of the change is that the German localization of 'XY Explorer' will use a customized button label and button tool tip (whereas all non-German localizations of 'XY Explorer still use the default label texts and tool tips). Furthermore, the entry for 'Undo' is removed from 'XY Explorer's glossary.

When customizing textual resources, there's no need to redefine all labels or all languages. Whenever you do not provide a customized version, the best matching generic version of the label will be used. That means, if a default version for the currently active language exists, the language specific default version, otherwise the `"en_US"` default version.

2

Data Import Modules

This part of the user's guide contains a reference documentation of all data import and data preprocessing modules of Synop Analyzer. Depending on your license, not all of the modules described here may be visible for you. You can activate those modules by updating your license.

Data Source Specification: The Data Source Specification module accesses, loads and transforms one or more input data sources and creates one single 'in memory' data object on which all data analysis modules can operate. Three types of data sources can be accessed directly: relational database tables, flat files and MS Access MDB files. Spreadsheets can be accessed via the Spreadsheet Import panel.

Spreadsheet Import panel: The Spreadsheet Import panel is a wizard which converts a complex spreadsheet (such as a MS-Excel document) into a flat data structure which is suitable for being imported into the Input Data panel.

Importing Data from Google Analytics: The Google Analytics import module reads web page analytics data via the Google Analytics API and opens them in the Input Data panel.

Data Transformations: The Record Grouping module starts from an existing input data source within Synop Analyzer and groups its records into a smaller number data groups, each group consisting of one or more records of the original data source. The new data are opened in a new Input Data panel tab on the left side of the Synop Analyzer workbench.

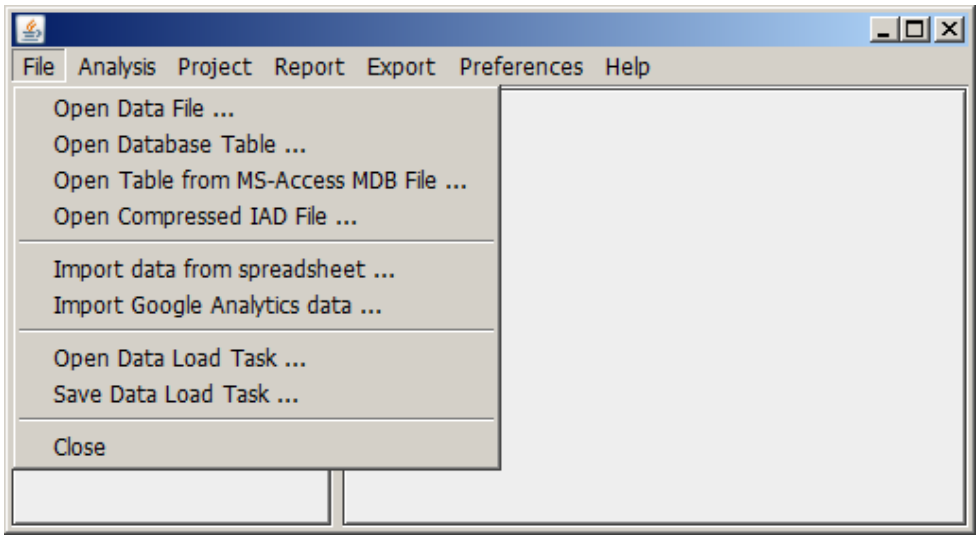
2.1 The Data Source Specification Panel

2.1.1 Supported data formats and data sources

Synop Analyzer is able to read data from the following data sources:

- Tables or views from all database management systems (DBMS) which support the JDBC data exchange interface.
- Microsoft Access© tables stored as MDB files.
- Spreadsheets in the Microsoft Excel© formats `.xlsx` and `.xls`. For importing data from spreadsheets with a complex structure of data, meta data, formula and textual explanation cells, see Importing data from spreadsheets.
- Flat text files in which the first row optionally contains the column names and the following rows the column values. The columns must be separated by a separator character such as `<TAB>`, `'|'`, `','`, `';'`, or `' '`.
- XML files. Here, the first repeatedly occurring XML tag in the hierarchy level directly below the document's root element is interpreted as the data container which contains the information of one data record. The field names of the data record are automatically detected from the attributes and the sub-tags of that data record tag..
- Data retrieved via a web API such as the Google Analytics API.
- Files in the Synop Analyzer-proprietary compressed `.iad` data format.

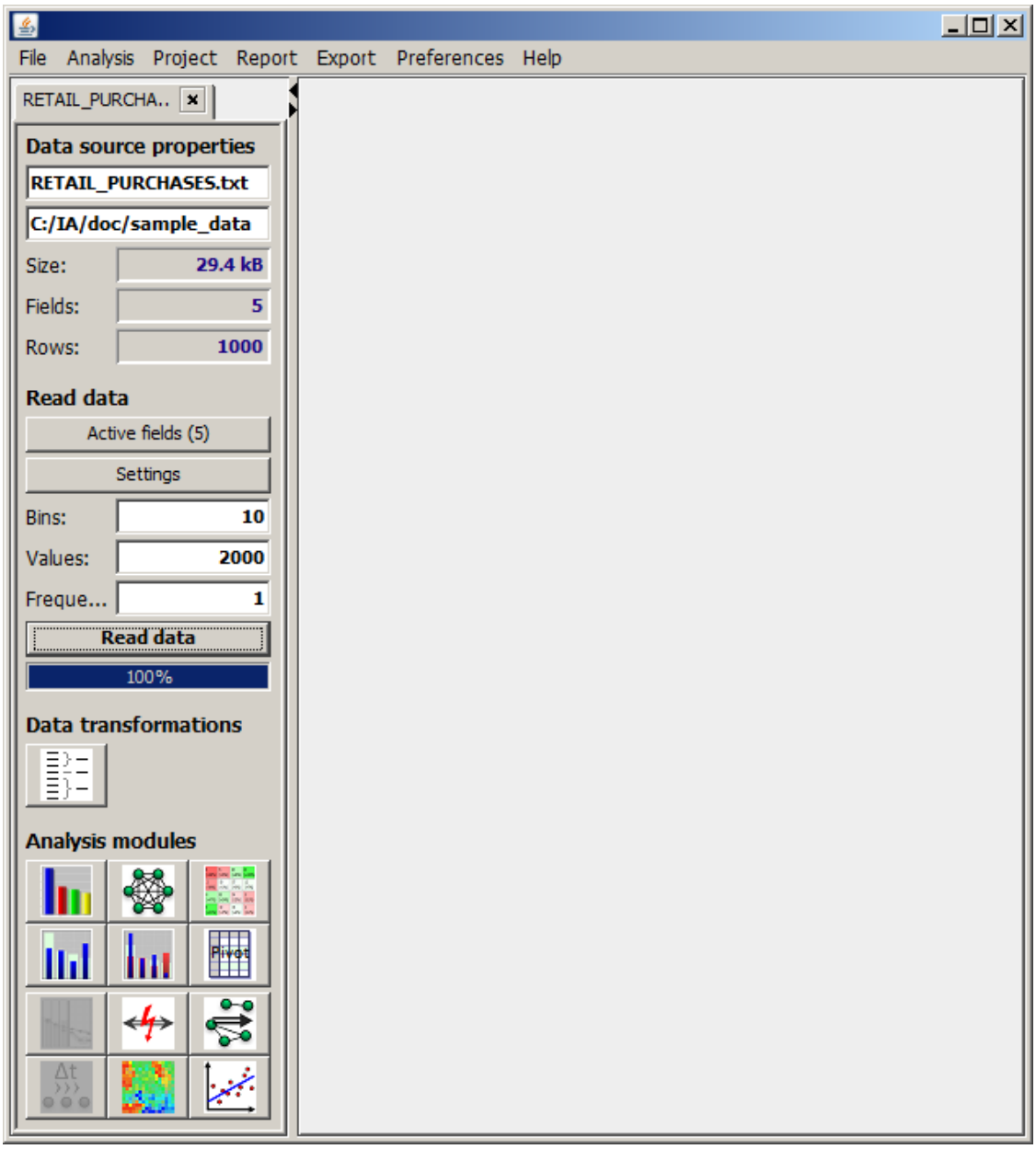
A new data source can be opened by clicking on the item **File** in Synop Analyzer's main menu:



Once you have opened a data source and you have specified some additional data import settings such as data field usage types, joins with auxiliary data, field discretizations, or computed fields, you can save these settings by clicking on **File** → **Save data load task** in Synop Analyzer's main menu. This will create a parameter file in XML format. You can later re-open this XML file via **File** → **Open data load task**.

2.1.2 The 'Input Data' panel

When a data source has been selected using the menu item **File**, an input data specification panel opens up in the left part of the Synop Analyzer screen. By pressing the **Start** button in the middle of that panel, the data are read into your computer's memory. Once this process is finished, the buttons in the lower part of the panel become enabled. Using these buttons, you can start Synop Analyzer's different data analysis and exploration modules.



When reading a data source, Synop Analyzer uses certain predefined settings and makes some assumptions as to the desired usage of the single data fields. The settings can be introspected and modified using the menu item **Preferences → Data Import Preferences**. Further assumptions and parameter settings are directly shown in the input data panel. Some basic parameters are visible in the upper part of the panel:

- **Active fields:**
  This button opens a dialog window in which active and inactive data fields can be selected and the roles of the active fields in the subsequent analysis steps can be specified. A more detailed description is given in The 'active fields' pop-up dialog.

- **Settings:**
  This button opens a dialog window in which active and inactive data fields can be

selected and the roles of the active fields in the subsequent analysis steps can be specified. A more detailed description is given in The 'settings' pop-up dialog.

- **Bins:**
  Several data exploration modules of Synop Analyzer display histogram charts of the data field's value distributions. For that purpose, the values of numeric data fields must often be discretized into a manageably small number of value ranges (intervals), otherwise the resulting histogram charts would become completely overcrowded. The number given in this parameter input field is the desired default number of histogram bars for all numeric data fields. The choice of the actual interval boundaries as well as the scaling - equidistant or logarithmic - is thereby left to a software heuristics. For single data fields, this behaviour can be overwritten, see User specified binnings and discretizations.

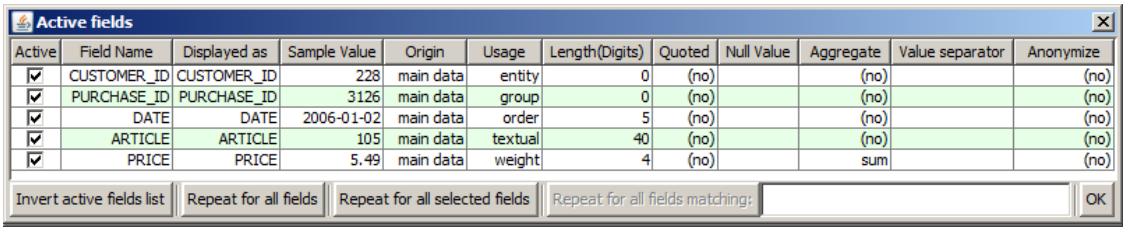
- **Values:**
  Several data exploration modules of Synop Analyzer display histogram charts of the data field's value distributions. For that purpose, the less frequent values of textual data fields must sometimes composed into groups, otherwise the resulting histogram charts would become completely overcrowded. The number given in this parameter input field is the desired maximum number of histogram bars for all non-numeric data fields. If the field contains more different values, the most frequent values get their own histogram bin, the remaining values are combined int one single value group called 'others'. For single data fields, this behaviour can be overwritten, see User specified binnings and discretizations.

- **Frequency:**
  This parameter defines a lower boundary for the number of data records or data groups on which a value of a non-numeric data field must occur for being tracked as a separate field value and a separate bar in histogram charts. Less frequent values will be grouped into the category 'others'.

Hint: You can duplicate a data import panel by right-clicking on the tab header of the panel. This creates a new input data panel in addition to the existing one. The new panel inherits all settings and specifications for importing the data that were specified on the original panel. You can now modify some of these settings in order to generate a second, slightly modified view on the data.

2.1.3 The 'active fields' pop-up dialog

Clicking on the button **Select active fields** in the input data panel opens a pop-up dialog in which all data fields of the current data source are displayed. The picture below shows the pop-up dialog for the sample data source `doc/sample_data/RETAIL_PURCHASES.txt`.



In the dialog, you can define several properties for each data field:

- **Active:**
  Deactivating this check box hides the data field when the data are read. The data source is treated as if the field was not present in the data.
- **Field name:**
  This table column is non-editable. It displays the original field name as it appear in the data source.
- **Displayed as:**
  In this table column you can define a new name which will be displayed instead of the original field name in all subsequent analysis results.
- **Sample value:**
  In this table column, the first value of the data field is displayed.
- **Origin:**
  This table column is non-editable. It displays the source of the data field. For data fields from the main data source, **main data** is displayed, **joined data** for data fields from auxiliary tables which were joined in, **computed** for computed fields and **replaced** for data fields which were present in the main data or an auxiliary data source but which were replaced by computed fields.
- **Usage:**
  In this table column you specify the data type and the usage mode of the corresponding data field. The default is **automatic**, which means that the field's data type is automatically set to **textual**, **Boolean**, **numeric** or **discrete (numeric)** based on an analysis of the field's first values, or, if the data source is a relational database, based on the field's data type in the database.
  This default handling can be modified by mouse-clicking into the table cell. On the one hand, one can manually specify the field's data type to be **textual**, **Boolean**, **numeric** or **discrete (numeric)**. Most practical relevance has the case that a data field with purely numeric values is to be treated as **textual** or **group** field, for example if the field is an ID or key field, for which the standard numeric field treatment, which involves calculating value ranges and statistics such as mean and standard deviations, would make no sense. In the screenshot above, the ID field **ARTICLE** has been treated in this way.
  On the other hand, you can specify four specific field usage modes which do not only define the field's data type but also its role and function within the data source. None of these four field usage modes may be set for more than one data field:
  Order denotes a data field whose values are time stamps, dates or another numeric

ordering criterion which specifies the time and order at which the other field values of the data record have been recorded. In the screenshot above, the field **DATE** has been specified as order field.
Weight specifies a numeric data field whose values contain the price, cost, weight, importance or another quantitative rating number which can be attributed to the corresponding data record. In the screenshot above, the field **PRICE** has been treated in this way.
Group should be used for a field which does not contain any independently usable information but serves for marking several adjacent data records as parts of one single data group. In the screenshot above, the field **PURCHASE_ID** has been marked as group field: it marks several consecutive data sets as parts of one single purchasing transaction. If a group field has been defined, all subsequently generated statistics and analysis results do not count and display data record numbers but data group numbers.
Entity denotes a second, higher-level grouping of data records on top of the group field. The specification of an entity field is particularly important for sequential patterns analyses. In this case, the group field defines groups of simultaneous events, the entity field defines 'entities' or 'subjects' to which time-ordered series of groups of simultaneous events can be attributed. Typical entity field - group field pairs are customerID and purchaseID, productID and productionStepID, or patientID and treatmentID.

- **Length (Digits):**
  For non-numeric fields, this is the maximum number of characters within a field's values; longer values will be truncated when reading the data into memory. For numeric fields, this is the numeric precision. For example, if this value is 4, then the number 1.2345 will be read-in as 1.235 because all digits after the fourth one will be rounded away.

- **Quoted:**
  In this table column the user can tell Synop Analyzer that some or all values of a data field are enclosed into single or double quotes in the data source and that these quotes should be ignored and stripped away when reading the field's values. Note: if all values of a field are enclosed in the same type of quotes, Synop Analyzer automatically recognizes and removes these quotes when reading the data.

- **Null value:**
  Per default, Synop Analyzer interprets the empty string and the value SQL NULL (when reading from relational databases) as 'no value available'. In real-world data, there are often additional special field values which indicate the absence of a valid value, for example the entry '-' in a name or address field, or the value '1900-01-01' in a data field, or '-1' in a field which should contain positive numbers. Those specific 'placeholder' values should be entered into the table column 'Null value' so that Synop Analyzer can correctly represent the intended purpose of these values.

- **Aggregate:**
  Once you have defined a group field, n consecutive data sets with identical group field values are treated as one single data group. For each nummeric data field, such

a group contains up to n different numeric values. The question now is: how should these single values be aggregated in order to form one single value which can be attributed to the data group? Per default, the sum of all values will be calculated. If you want to define another aggregation method, for example the mean, minimum, maximum, spread (maximum minus minimum), relative spread (maximum minus minimum dividey by mean), count (the number of valid values), or existence (1 if a valid value exists, 0 otherwise), click on the table cell **Aggregate** and selecte the desired aggregation mode. In the screenshot above, the aggregation function of the weight field **PRICE** has been set to **sum**, and consequently, the field has been renamed to 'PurchaseValue' as the field now represents the total value of each purchase.
There are more complex aggregation methods which involve the values of other data fields as aggregation criteria. The method **Value at which XXX is maximum**, for example, selects the field value of that data record within the data group at which the data field `XXX` assumes its maximum value within the data group.

- **Value separator:**
  In this column you should enter a value if the corresponding data field contains set-valued entries, for example entries of the kind `cream;diapers;mineral water;baby food`. In these cases, the software must be told that the entry should not be treated as one single information but as a set of several single bits of information, that means in our example the set of the four values `cream`, `diapers`, `mineral water` and `baby food`. In order to achieve this, enter the separator character into the **Value separator** column, in our example the character `;`. Once a separator character has been defined, eventually present braces around the entire expression are automatically identified as set indicators and ignored when extracting the single values of the set..

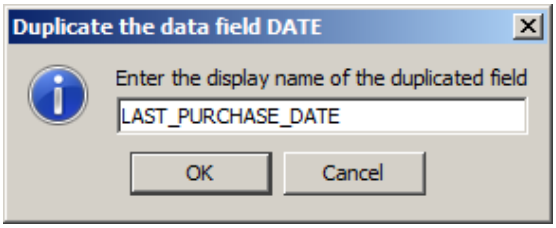
- **Anonymize:**
  Sometimes analysis results created on confidential data are to be distributed to a larger receiver group which is not authorized to see all parts of the data. For this case, Synop Analyzer offers the possibility to anonymize certain field names and/or these fields' values before reading the data and creating analysis results. For each data field, the anonymization level can be set individually. There are three modes: one can anonymize the field name, the field's values or both. If you permanently store the imported data as a compressed `.iad` file, this file only contains the anonymized data and not the original values. Hence, you can distribute the `.iad` file together with the analysis results.

Duplicating data fields

Sometimes it is desirable to use one single data field from the original data in two different ways in Synop Analyzer. For example, you could use the time stamp of a transaction within a transactions data collection both as grouping criterion (usage 'group') and as time order field (usage 'order'); or you could use and display one single date field with the

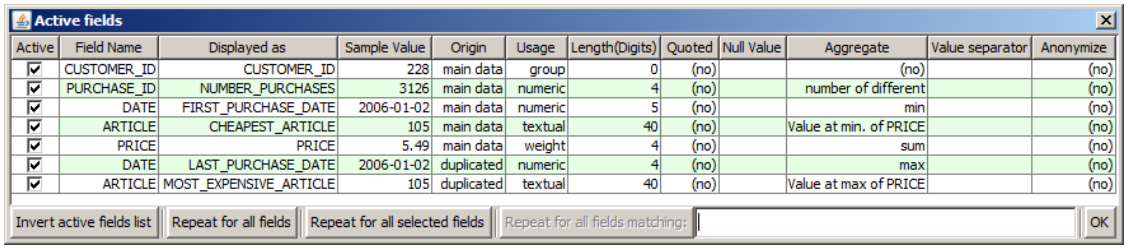
two different aggregation modes 'minimum' and 'maximum' in order to show the date of the first and the last transaction of a customer.

In Synop Analyzer, you can duplicate a data field by right mouse click on the table row representing the data field in the pop-up dialog **Select active fields**. In a second pop-up dialog you will then be asked to specify the name of the new copy of the original data field. Make sure that the new display name is unique:



After closing the field name dialog the duplicated data field will appear as a new row in the table of all available data fields. You can then specify the desired usage type, aggregation mode and other desired properties of the duplicated field.

The screenshot below shows a practical example in which defining duplicated fields is very helpful: on the sample data `sample_data/RETAIL_PURCHASES.txt` we have defined the field `CLIENT_ID` as the grouping criterion. The field `DATE` has been duplicated, renamed into `FIRST_PURCHASE_DATE` and `LAST_PURRCHASE_DATE` and furnished with the aggregation modes 'minimum' resp. 'maximum'. Similarly, the field `ARTICLE` has been duplicated, renamed into `CHEAPEST_ARTICLE` and `MOST_EXPENSIVE_ARTICLE` and furnished with the aggregation modes 'value at which `PRICE` is minimum' resp. 'value at which `PRICE` us maximum'. wurden dupliziert, umbenannt in `ERSTES_KAUFDATUM` und `LETZTES_KAUFDATUM` und mit der jeweils passenden Aggregierungsfunktion (Minimum bzw. Maximum) versehen. Schließlich wurde noch das Datenfeld `ARTIKEL` dupliziert, umbenannt in `BILLIGSTER_ARTIKEL` und `TEUERSTER_ARTIKEL` und mit der Aggregierung 'Wert, bei dem `PREIS` minimal ist' bzw. 'Wert, bei dem `PREIS`maximal ist' versehen.


Active fields
Active | Field Name | Displayed as | Sample Value | Origin | Usage | Length(Digits) | Quoted | Null Value | Aggregate | Value separator | Anonymize
CUSTOMER_ID | CUSTOMER_ID | 228 | main data | group | 0 | (no) | (no)
PURCHASE_ID | NUMBER_PURCHASES | 3126 | main data | numeric | 4 | (no) | number of different | (no)
DATE | FIRST_PURCHASE_DATE | 2006-01-02 | main data | numeric | 5 | (no) | min | (no)
ARTICLE | CHEAPEST_ARTICLE | 105 | main data | textual | 40 | (no) | Value at min. of PRICE | (no)
PRICE | PRICE | 5.49 | main data | weight | 4 | (no) | sum | (no)
DATE | LAST_PURCHASE_DATE | 2006-01-02 | duplicated | numeric | 4 | (no) | max | (no)
ARTICLE | MOST_EXPENSIVE_ARTICLE | 105 | duplicated | textual | 40 | (no) | Value at max of PRICE | (no)
Invert active fields list
Repeat for all fields
Repeat for all selected fields
Repeat for all fields matching:
OK

When you import and display the data in Synop Analyzer in this way, the displayed data contain one single row per customer. The data row contains the ID of the customer, his or her total number of purchases, the total amount of money spent so far, the cheapest and the most expensive article purchased so far, and the date of the first and the last purchase.

Hints for minimizing memory requirements and for maximizing processing speed

- Textual data fields with many (more than ca. 5000) different values have high memory requirements and reduce the speed of following analysis steps. Therefore, free text fields and ID or key fields should be deactivated in the 'select active fields' dialog whenever possible.
- Sometimes you still want to keep a key field with tens of thousands or even millions of different values, for example because your analysis aims at creating small subsets of the original data and in these data subsets you need the key attribute for unambiguously identifying each selected data record. The selection of a target group of customers for a marketing campaign is such an example: here, you want to keep the 'customerID' field even if it contains millions of different IDs. In this case you should mark the data field as **group** field and not as **textual** field: internally, the treatment and memory storage model of group fields is optimized for many different values, the treatment of textual fields is not.
- For numeric data fields with many different values, the memory requirements for storing them heavily depends on the numeric precision with which the field values are read-in. Such a data field, when read-in with a numeric precision of 7, can consume up to 1000 times more memory than the same data field read-in with a precision of 4. A precision of more than 3 to 5 digits is rarely needed for analysis and data mining tasks. Therefore, on large data you should reduce the numeric precision to the minimum acceptable number.

2.1.4 The 'Settings' pop-up dialog

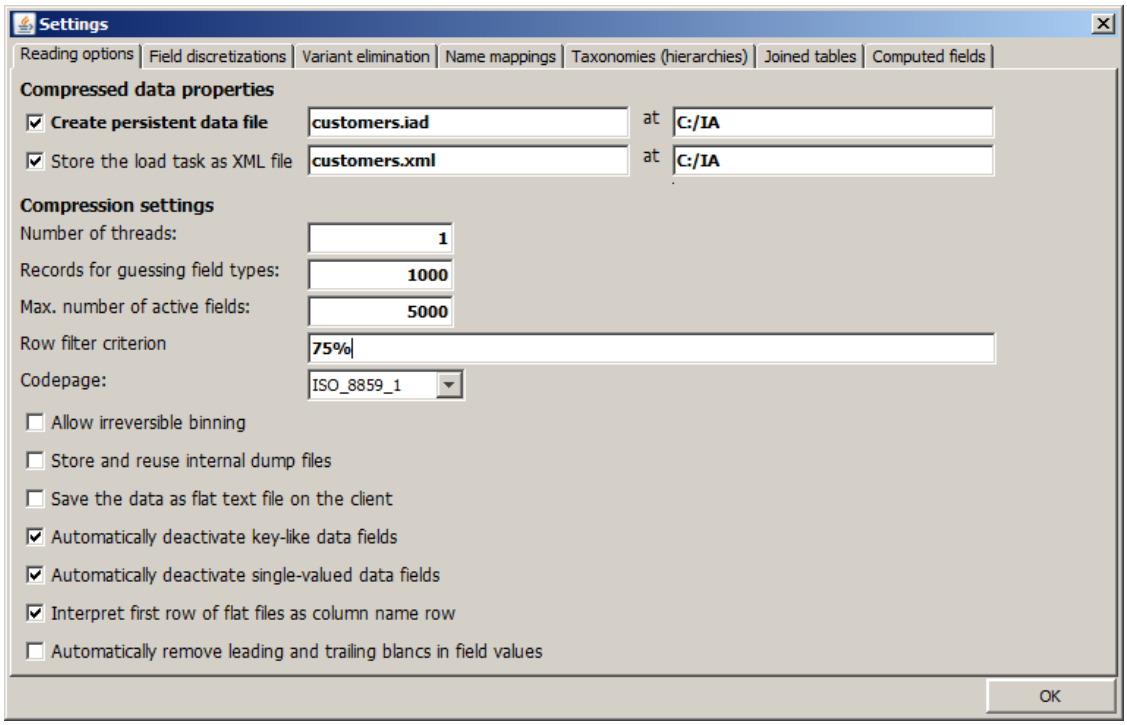
Many more advanced options for customizing the data preparation and data importing process are accessible via the pop-up panel **Settings**. The panel is organized into seven 'tabs' or pages, which will be described one by one in the following sections of this document.

The 'Reading Options' tab

In the upper part of the first tab named **Reading options**, one can specify whether the binary data object which has been composed in the computer's main memory is also stored permanently in the form of an `.iad` file. The `iad` data format is an Synop Analyzer-proprietary data format; it contains a compressed binary representation of the input data as well as all data preprocessing and data joining steps defined in the input data panel. An `iad`can be loaded from disk very quickly - much faster than the time it took to read the data from the original data sources.

If only the data preparation and data import settings but not the imported data themselves are to be stores, one can activate the check box **Store the load task as XML**

file. This option has the advantage that one can repeatedly re-load the must current data snapshot from the original data source without to be obliged to re-enter the data preparation and data import settings.



In the lower part of the tab you can modify various predefined settings for the process of reading the data into the computer's memory:

- **Number of threads:**
  Specify an upper limit for the number of parallel threads used for reading and compressing the data. If no number or a number smaller than 1 is given here, the maximum available number of CPU cores will be used in parallel.
- **Records for guessing field types:**
  When reading input data from flat files or spreadsheets, the data source does not provide meta data information on the types of data (integer, Boolean, floating point, textual) to be expected in the available data columns. Therefore, a presumable data type has to be derived from looking at the data fields actual content. The parameter 'number of records for guessing field types' determines, how many leading data rows are read from the data source for guessing data field types.
- **Max. number of active fields:**
  If a data source contains a large number of data fields, it is helpful for the clarity and speed of any subsequent analysis steps to concentrate on not more than 40 to 50 of the data fields. By entering a number smaller than the current number of active data fields, you ask the software to find a subset of all data fields which contains as much as possible of the information contained in the entire data. This will be reached by deactivating fields with many missing values, almost unique-valued fields

or almost single-valued fields, and by dropping all but one field from each tupel of highly correlated fields such as AGE and DATE_OF_BIRTH.

- **Row filter criterion:**
  Here, you can define a data row filter criterion. This can be specified in the form of a percentage; for example, the filter `5%` means that a random sample of about 5% of the entire data will be drawn. The filter `!5%` creates the inverse sample which contains exactly those data records which are not part of the `5%` sample. If your data source is a database table, you can also submit the filter criterion in the form of a SQL WHERE clause; for example, `WHERE AGE<40` means that only those data sets are to be read in which the field `AGE` has a value of less than 40.

- **Codepage:**
  The codepage (encoding scheme) in which the input data are encoded defines the way in which bits and bytes in the source data are interpreted as letters and symbols. Synop Analyzer's standard codepage is the US and Western European default ISO_-8859_1, which is a 1-byte codepage (1 byte per character). Another frequently used code page is UTF-16 (2 or more bytes per character).

- **Allow irreversible binning:**
  If this check box is marked, numeric data fields can be discretized into a small number of intervals, and the original field values are irreversibly replaced by interval indices. For example, the value `AGE=37` might be replaced by `AGE=[30..40[`, and in the compressed data, the precise value 37 will be irreversibly lost. This discretization can significantly reduce the memory requirements of the data.

- **Store and reuse internal dump files:**
  When reading data from flat files or database tables, a temporary buffer object is created for each data field. Storing and reusing these temporary objects can considerably speed up the data reading process in subsequent data reading steps from the same data source.

- **Save the data as flat text on the client:**
  When reading data from a remote text file or database table, copy the data in the form of a flat text file into the current working directory on the local machine. This can speed up subsequent data reading steps if the bandwidth to the remote data server is limited.

- **Automatically suppress key-like fields:**
  Key-like fields are non-numeric data fields in which (almost) every data record has a unique value. This option automatically sets the usage mode of all those data fields which have not been specified as group field to 'inactive'.

- **Automatically suppress single-valued fields:**
  Single-valued fields are data fields in which (almost) every data record has the same value. If this checkbox has been marked, the usage mode of all those data fields is automatically set to 'inactive'.

- **Interpret first row of flat files as column name row:**
  Per default, the first row of a flat text file will be interpreted as head row containing the column names. You should deactivate this option when reading flat files which do not have a head row.

- **Automatically remove leading and trailing blancs in field values:**
  f this option is activated, leading and trailing spaces are automatically removed when importing the data.

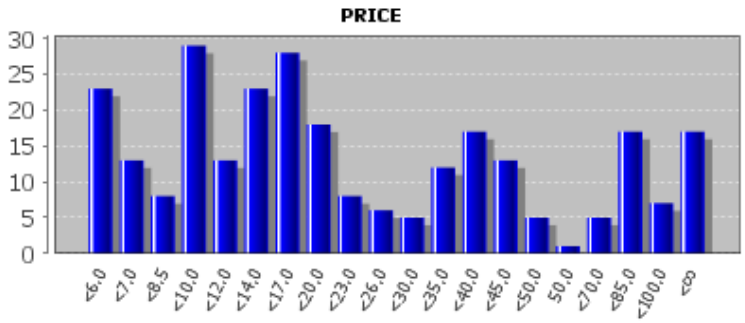
2.1.5 User specified binnings and discretizations

This tab provides the means for a field-specific modification of the default settings of

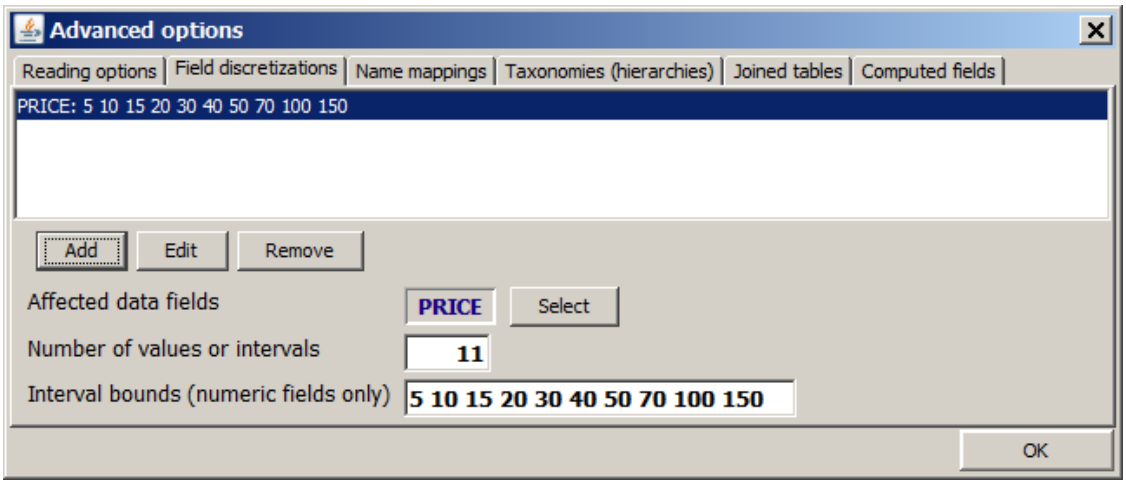
- how many different field values of non-numeric fields are treated as separate values and which values are grouped into 'others'

- how many value ranges (intervals) are used to display the value distributions of numeric fields in histogram charts and what are the interval boundaries.

In the following, we want to demonstrate this using the sample data `RETAIL_PURCHASES.txt`. If these data are imported into Synop Analyzer as described in the 'active fields' pop-up dialog, with `PRICE` as weight field and `PURCHASE_ID` as group field, then the values of the field `PRICE` will be partitioned into the following ranges:



This range partition shall now be replaced by the 11 ranges with the boundaries 5, 10, 15, 20, 30, 40, 50, 70, 100 and 150. We open the tab **Field discretizations** and enter the following string into the field **Interval bounds (numeric fields only)**: `5 10 15 20 30 40 50 70 100 150`. Then we finish the specification by pressing <TAB> or <Enter> and press **Add**. The tab now looks like this:



Each series of interval boundaries such as `5 10 15 20 30 40 50 70 100 150` which has been specified for a field discretization is stored in a 'interval boundaries history' store in the preference settings of Synop Analyzer. You can access the the 50 most recently used interval boundary sets from the pull-down selection box at the right side of the input field for interval boundaries.

If we close the pop-up dialog now using **OK** and re-open an analysis view which shows value distribution histograms, the histogram for the field `PRICE` shows the desired value ranges:

You can delete manually defined value range definitions by means of **Delete** and modify them using **Edit**.

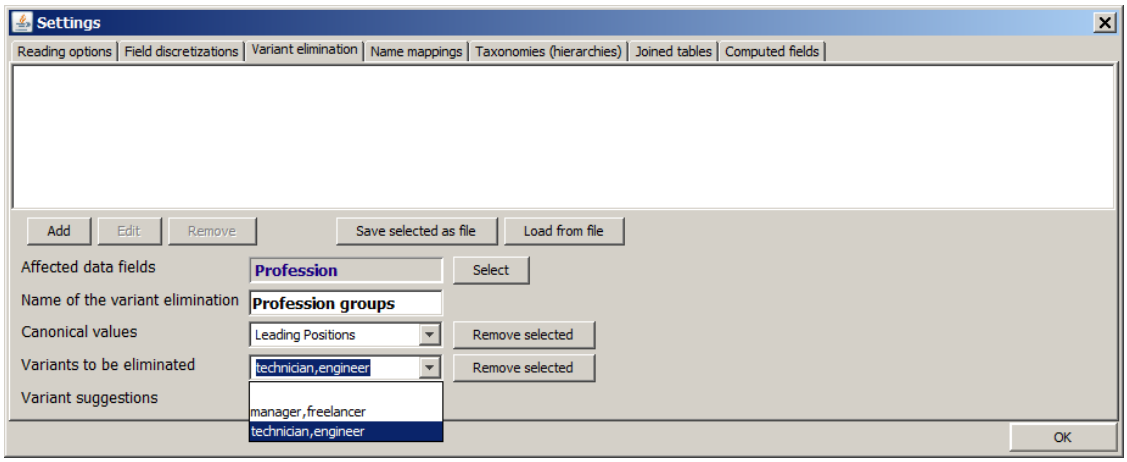
2.1.6 Value groupings and variant elimination

This tab provides the means for reducing the set of field values of a textual data field. The most important application areas are:

- If the data contain misspelled entries or different names for identical things, for example the variants `TOYOTA COROLLA`, `Corolla`, `Corrola`, `Toyota Corolla GT 2.0`, `T.Cor.` für das car mark `Toyota Corolla`.

- If the information contained in the data field are too fine-grained and ought to be summarized into a smaller number of groups or categories, for example the professions supermarket articles `apples Granny Smith`, `apples Golden Delicious`, `apples Braeburn` and `apples Idared` to the group `apples`.

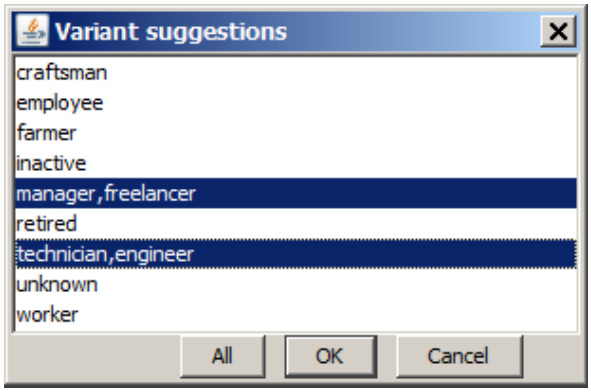
In Synop Analyzer, you can define several value groupings or variant eliminations within the tab **Variant elimination**, and each of them can be activated for one or more data fields. Per default, all variant eliminations defined when importing a certain input data source are stored as a part of the 'data load task', that means the XML file which stores all settings and user-defined specifications which have been performed for the input data source. But using the button **Save selected as file** you can also save a variant elimination as a data-independent persistent XML file. This file can later be loaded and activated for a new data source using the button **Load from file**. This mechanism enables the creation of a data-independent 'knowledge base' of spelling variants for specific application areas, for example a data-independent knowledge base of Toyota car names.



Each variant elimination consists of the following parts:

- The data fields on which the variant elimination is to be applied. In the example shown above, which is built on the sample data `doc/sample_data/customers.txt`, the variant elimination shall be applied to one single field, `Profession`.
- A unique name for the variant elimination. In the example shown above, we use the name `Profession groups`.
- The specification of at least one 'canonical form' or group value. In the example shown above, we have defined one single group value called `Leading Positions`.
- The definition of several variants or at least one variant pattern for each value group. In the example shown above, we wanted to combine the two values `manager,freelancer` and `technician,engineer` into the new group value `Leading Positions`. The variant specifications could also contain regular expressions; for example, the character `*` stands for 0, 1 or more arbitrary characters; the expression `[Aa]` stands for 'either A or a', the expression `[a-z]` for exactly one lower case letter.In order to make it easier to work with this feature for users which are not

familiar with regular expressions, Synop Analyzer interprets each appearance of '*' as a general wildcard representing zero or more arbitrary characters. That means, the expression `Tech*` is interpreted as 'All strings starting with `Tech`'; in correct regular expression syntax we would have to write that as as `Tech.*`, which is also possible in Synop Analyzer.
The variants can either be typed in one by one using the input field **Variants to be eliminated**, or one can select the desired values from an lexically sorted list of all different field values of the affected data field which is opened by pressing the button **Variant suggestions**. However, this latter way is only available if the input data have been read in by Synop Analyzer before using the button **Read data** in the left column of the main screen. In our example shown above, the button **Variant suggestions** would show us, if we have read in the data `customers.txt` before, the following list, from which we can select the desired values by pressing the **OK** button:



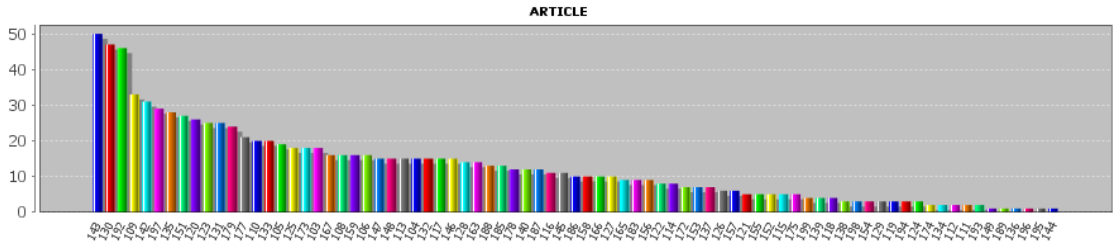
- Finally we have to specify whether the original data field values are to be replaced by the value group names matching them, or whether they are to remain in the data in addition to the newly added value group names. This is done by means of the check box **Keep also the original field values**.

Once defined, the variant elimination settings will be applied to all subsequent data reading processes on the current input data. In our example, the two values `manager,freelancer` and `technician,engineer` of the data field `Profession` are replaced by the group value `Leading Positions`:

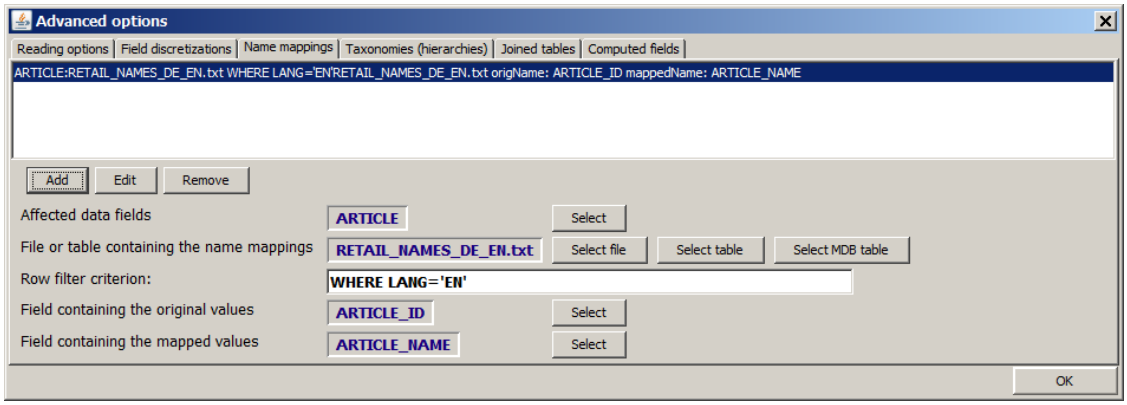
2.1.7 Name mappings

This tab provides the means for assigning more cleary understandable alias names to the values of a textual data field. These names can be read from an auxiliary file or database table which must contain at least two columns: one column must contain values which exactly correspond to the existing values of the textual field, the second column must contain the desired alias names for these values.

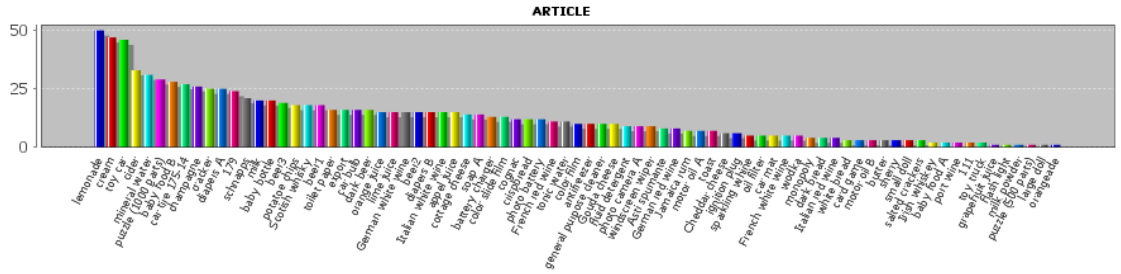
In the following, we want to demonstrate this using the sample data `RETAIL_PURCHASES.txt`. If these data are imported into Synop Analyzer as described in the 'active fields' pop-up dialog then the field `ARTICLE` contains hardly understandable 3-digit ID numbers. We would like to replace these numbers by textual article names.



A list of English and German article names is available in form of the file `doc/sample_data/RETAIL_NAMES_DE_EN.txt`. This file contains three columns: `ARTICLE_ID`, `ARTICLE_NAME` and `LANG`. `ARTICLE_ID` contains the same article identifier number which occur in the main data and `LANG` contains the two language identifiers `DE` and `EN`. We open the tab **Name mappings** within the **Advanced options** pop-up window and insert the entries shown in the picture below in the lower gray part of the tab. Then we press the **Add** button. The tab should look like this now:



If we close the pop-up dialog now using **OK** and re-open an analysis view which shows value distribution histograms, the histogram for the field `ARTICLE` shows the desired textual values:

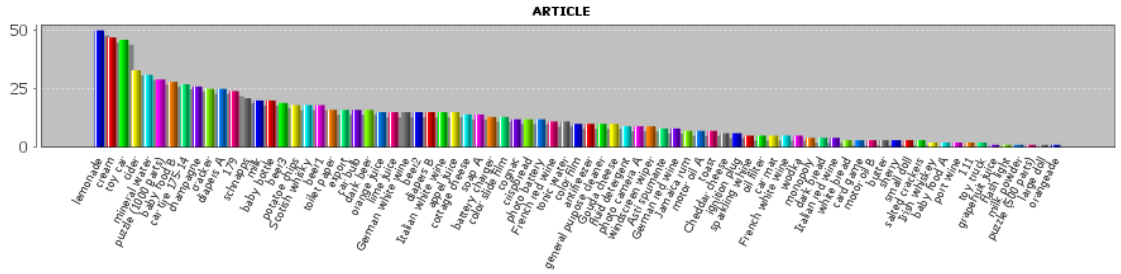


You can delete manually defined name mapping definitions by means of **Delete** and modify them using **Edit**.

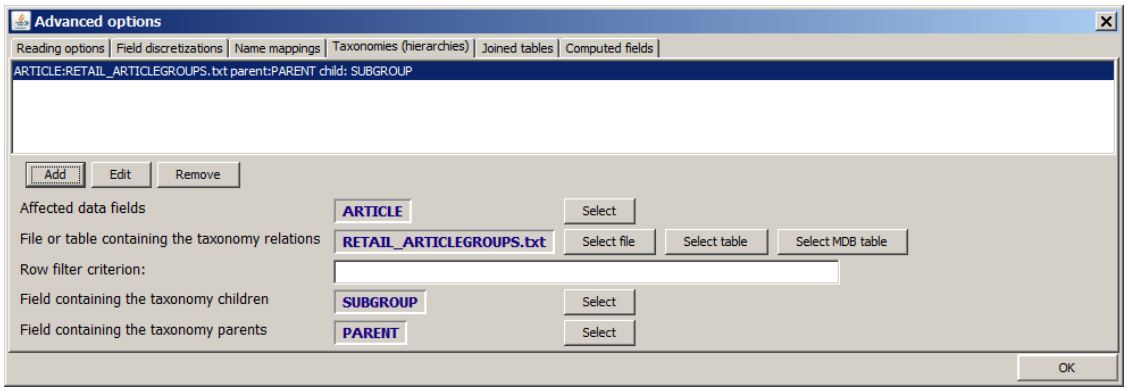
2.1.8 Taxonomies (hierarchies)

This tab provides the means for adding hierarchical grouping information to the values of a textual data field. Hierarchies, also called 'taxonomies', can be read from an auxiliary file or database table which must contain at least two columns: one column must contain the lower-level ('child') part of a hierarchy relation, the other column the higher-level ('parent') part.

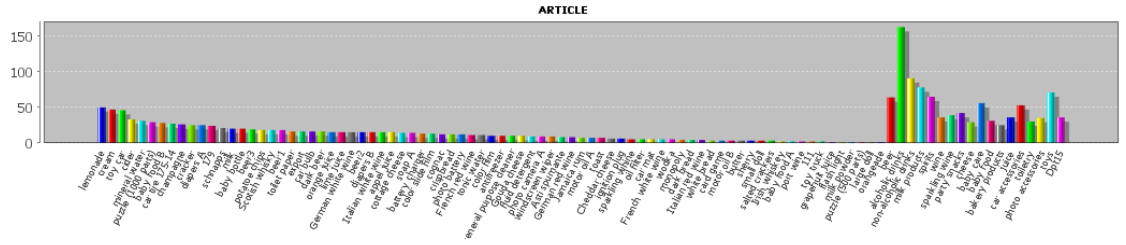
In the following, we want to demonstrate this using the sample data `RETAIL_PURCHASES.txt`. We assume that these data have been imported into Synop Analyzer and enriched with name mapping information as described in section Name mappings. We would like to add article group and article department information to the article names.



A list of article group and department information is available in form of the file `doc/sample_data/RETAIL_ARTICLEGROUPS_DE_EN.txt`. This file contains the columns: `PARENT` and `SUBGROUP`, which are easily identified as parent and child column. We open the tab **Taxonomies** within the **Advanced options** pop-up window and insert the entries shown in the picture below in the lower gray part of the tab. Then we press the **Add** button. The tab should look like this now:



If we close the pop-up dialog now using **OK** and re-open an analysis view which shows value distribution histograms, the histogram for the field `ARTICLE` shows new article group values and department values in addition to the existing article names. (Per default, the Synop Analyzer histograms show only up to 80 histogram bars. You can increase this value to 100 in the pop-up dialogs **Preferences** → **Univariate Preferences** and **Preferences** → **Multivariate Preferences** in order to obtain the result shown below.)



You can delete manually defined taxonomy definitions by means of **Delete** and modify them using **Edit**.

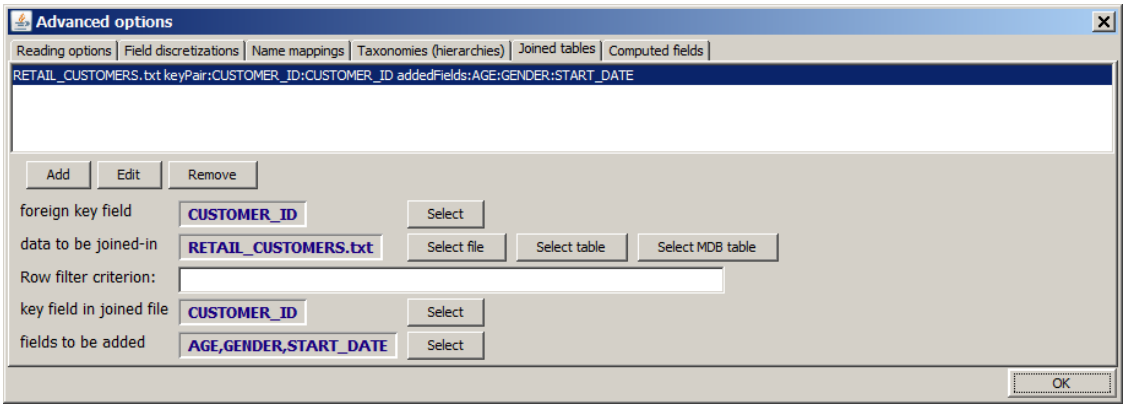
2.1.9 Joining with auxiliary tables

This tab provides the means for appending new data fields (columns) to an existing data source which has been opened in Synop Analyzer. The values in the new fields are obtained from a second data source; they are merged into the main data source via a foreign key - primary key relation between a field in the main data source and a field in the second data source. That means, the main data source must contain a data field ('foreign key field') whose values are the values of a primary key field in the second data source. It is not neccessary that the primary key field has been explicitly specified as primary key field within the second data source, it is sufficient that the field is a 'de-facto' key field in the sense that no value of the field occurs identically in more than one data row.

In the following, we want to demonstrate this using the sample data `RETAIL_PURCHASES.txt`. We assume that these data have been imported into Synop Analyzer and enriched with name mapping information as described in section Name mappings. We would like to add customer master data to these data. A master data file is available in

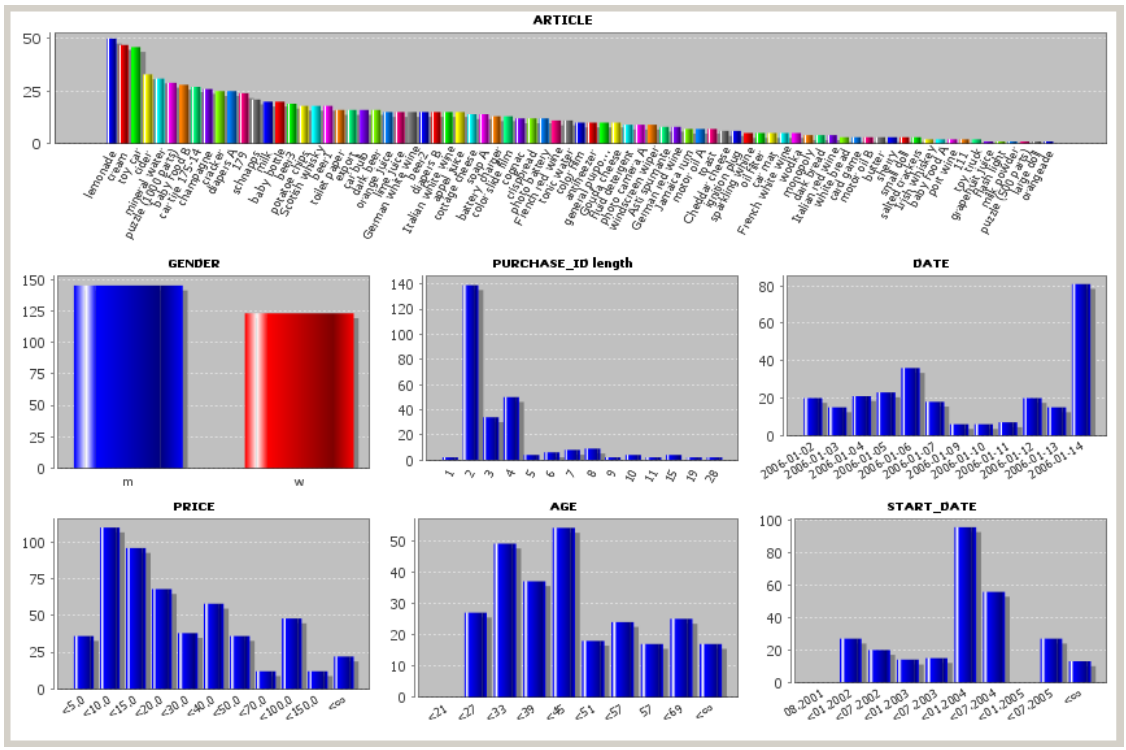
doc/sample_data/RETAUL_CUSTOMERS.txt. It contains the data fields AGE, GENDER and START_DATE (the latter being the date at which the customer loyalty card was handed out). The connection to the main data source is established via the foreign key - primary key pair CUSTOMER_ID (foreign key field in RETAIL_PURCHASES.txt) and CUSTOMER_ID (primary key field in RETAIL_CUSTOMERS.txt).

We open the tab **Joined tables** within the **Advanced options** pop-up window and insert the entries shown in the picture below in the lower gray part of the tab. Then we press the **Add** button. The tab should look like this now:



Note: the input field **Row filter criterion** can be used to make a field which is not a primary key field in the auxiliary data source behave like a primary key field. Imagine that we want to add a customer address field to the main data, but in the address master data there are customers for which we have two or more addresses, labeled by an address counter field ADDRES_NBR which contains a running number 1, 2, 3, etc.. In this form, we cannot join-in address information because for some customers we don't know which address to take. However, if we enter WHERE ADDRESS_NBR=1 into the field **Row filter criterion**, the address becomes unique and the joining-in canbe performed.

If we close the pop-up dialog now using **OK**, re-read the data and open an analysis view which shows value distribution histograms, three new histograms for the fields AGE, GENDER and START_DATE appear. These new fields can now be used just as if they had been present in the main data source right from the beginning. And if you persistently save the data as an .iad, the saved data also contains the three new fields.



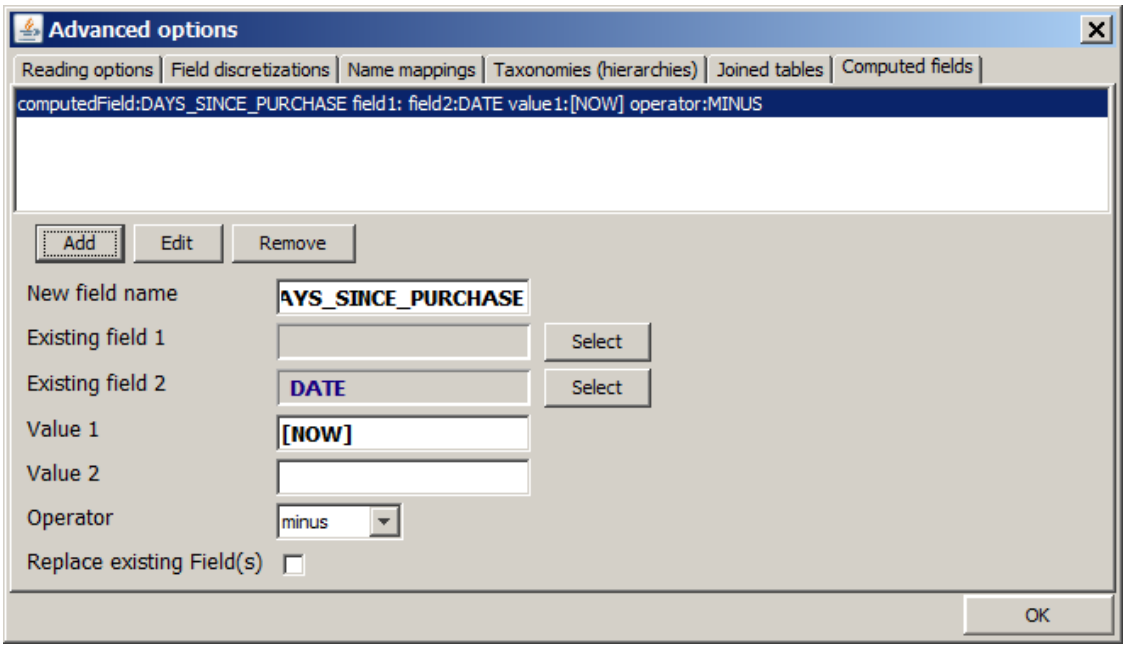
You can delete manually defined joined data definitions by means of **Delete** and modify them using **Edit**.

2.1.10 Computed data fields

This tab provides the means for appending new fields (columns) to an existing data source whose values are the result of applying a computation formula to the values of one or more existing data fields.

In the following, we want to demonstrate this using the sample data `RETAIL_PURCHASES.txt`. We assume that these data have been imported into Synop Analyzer and enriched with name mapping information as described in section Name mappings. We would like to add a new data field which contains the number of elapsed days between the date of the purchase and the current day at which the data analysis takes place. This information can be calculated from the value of the data field `DATE` and the current date.

We open the tab **Computed fields** within the **Advanced options** pop-up window and insert the entries shown in the picture below in the lower gray part of the tab. Then we press the **Add** button. The tab should look like this now:

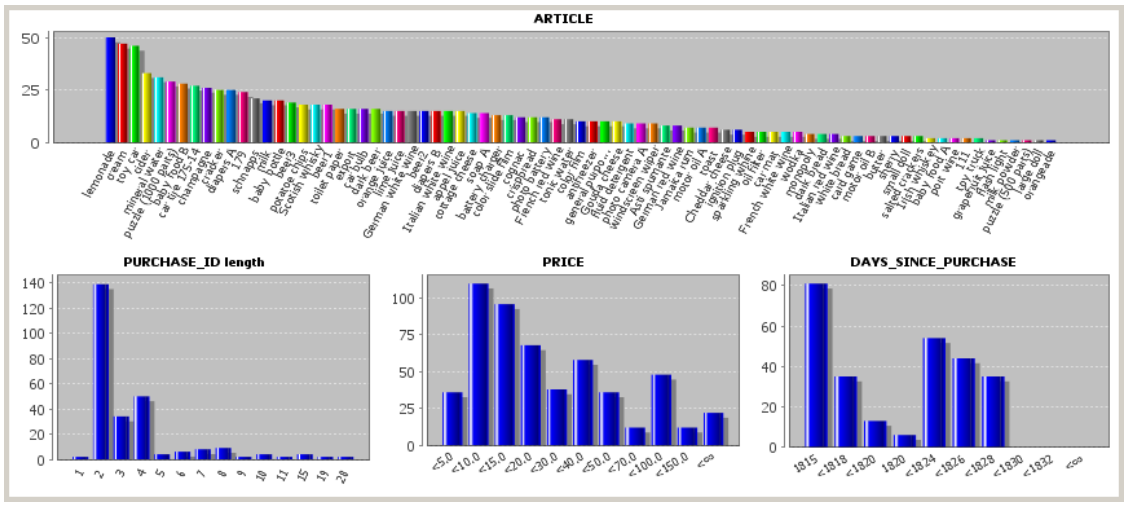


Note: In order to tell the software that the first part of the formula does not depend on the value of a data field, we have clicked in the button **Select** next to **Existing field 1**, then we have selected the last, empty entry in the pop-up menu. Without this step, the **Add** remains grayed-out and unusable.

Note: By activating the check box **Replace existing fields** you can tell the software that the existing fields involved in the formula should be removed from the data after the computation. However, this is not possible if one of the fields to be removed has been assigned one of the four special roles 'group', 'entity', 'weight' or 'order'. Therefore, the field `DATE` can not be replaced. One would get an error message if one tried to do so.

Note: The special constant `[NOW]` is a placeholder for inserting the current date and time of the time the data are read into memory.

If we close the pop-up dialog now using **OK**, read the data and open an analysis view which shows value distribution histograms, we see the new data field `DAYS_SINCE_PURCHASE`. In the screenshot below, we have deactivated the existing field `DATE` using the **Visible fields** button. Furthermore, we have increased the default number of histogram bars for numeric fields from 10 to 12 (using input field **#bins (numeric fields)**) before reading the data.



You can delete manually defined computed field definitions by means of **Delete** and modify them using **Edit**.

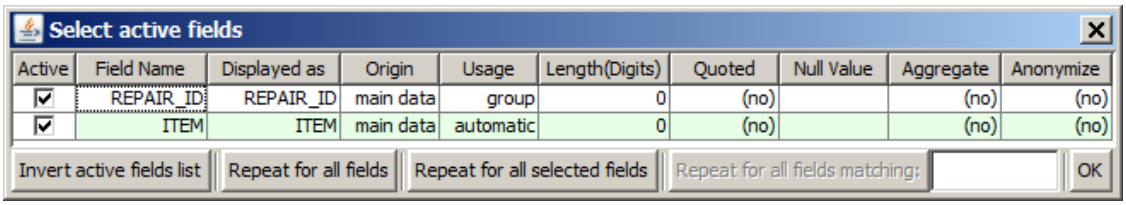
2.1.11 Transactional and streaming data

Automatically recorded mass data from logging systems - for example supermarket cash desk data, web stream data or server log data - often have only two columns: a counter or time stamp column and another column which contains all the information recorded at a certain counter state or time. The sample data file `doc/sample_data/CAR_REPAIR.txt` is an example for such a type of data. The file contains car read-out data which were recorded when a cars were connected to a testing device at a car repair shop.

```
REPAIR_ID ITEM
08760257  FINDING=ZHP3961323
08760257  ERROR_LOG=XZH60825
08760257  ERROR_LOG=XZH60820
08760257  ERROR_LOG=XZH60569
08760257  ERROR_LOG=XZH60565
08760257  FINDING=XXX3962357
08760257  ERROR_LOG=W37N422980723
08760257  ERROR_LOG=WCX2975749
08760257  FINDING=VPKN39674741
08760257  ERROR_LOG=VNU2975389
```

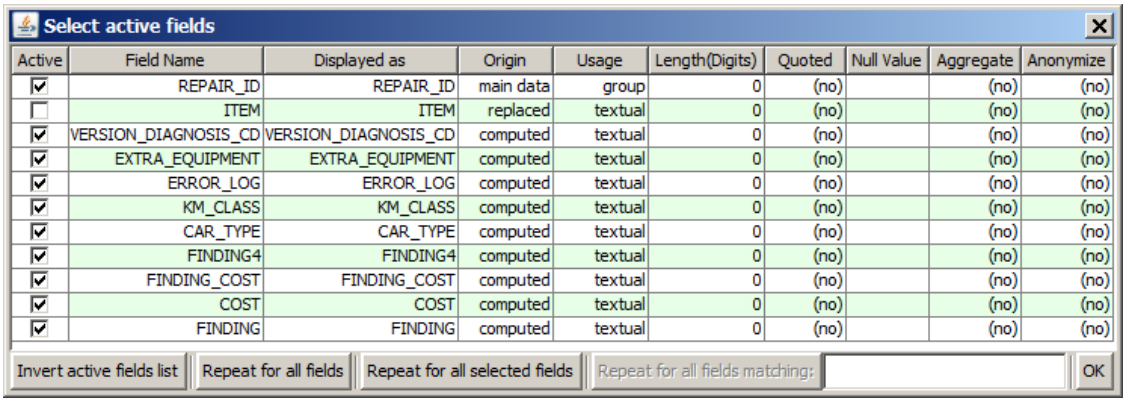
As can be seen from the data extract shown above, the second column contains the read-out information in the form `attribute name` = `attribute value`. The first column is the ID column. Its values indicate which data rows belong to one single car read-out.

When the data `CAR_REPAIR.txt` are read into Synop Analyzer, the field `REPAIR_ID` should be specified as group field. See section The 'active fields' pop-up dialog for more details.

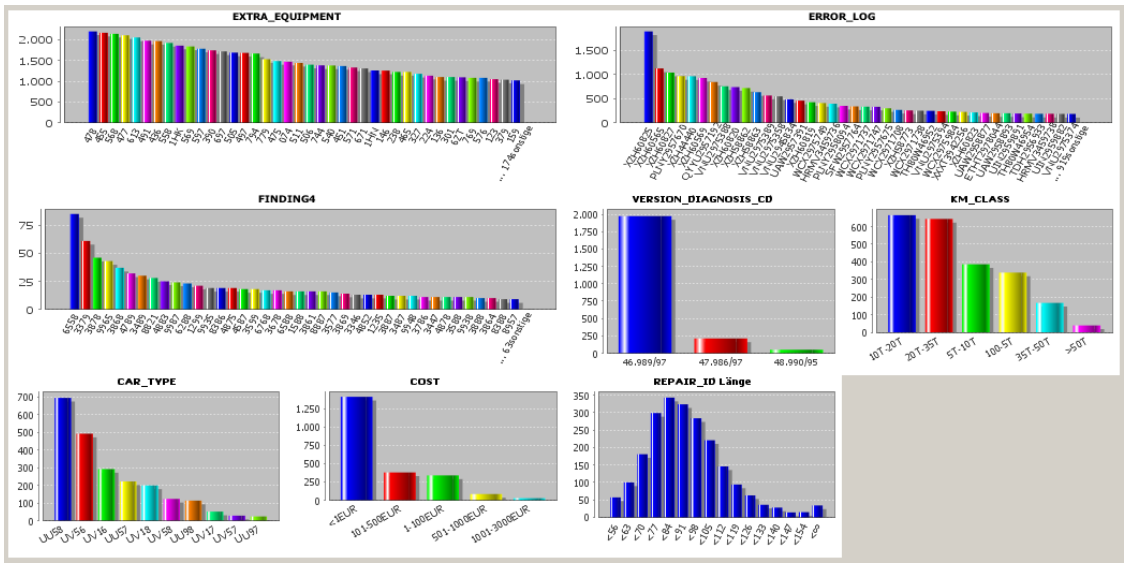


Whenever Synop Analyzer is reading data which contain - apart from a possibly specified group, entity, order and/or weight field - only one single, textual data field, the software checks whether it is able to detect internal structures and groups of information within the single textual field. In particular, Synop Analyzer searches for prefixes of the kind `attribute name=`, with the aim of identifying several different such prefixes and using them for information grouping.

Once the data have been read-in by pressing the button **Start** in the input data panel, one can introspect the different 'information groups' or prefixes by clicking on the **Select active fields** button: each prefix group is shown as a separate data field. In the example of the `CAR_REPAIR.txt` data, 9 such artificial data attributes are detected. The original data field `ITEM` has been marked as 'replaced':



In following data analysis steps, one can work with the artificial data fields as if they were full blown data fields from the data source. As an example, we show a univariate statistics view in which the value distributions of the artificial data fields can be studied:



At the end of this section we want to discuss the possible question what the advantage of the two-column 'transactional' data format is. The answer is that this 'slim' data format offers a very flexible possibility to store both set-valued and scalar-valued data attributes in one single flat data structure without any redundancy. In our example, some of the attributes typically contain many different values per repair ID, for example `ERROR_LOG`, `EXTRA_EQUIPMENT` or `FINDING`. Other attributes such as `CAR_TYPE` or `KM_CLASS` contain only one value per repair ID. The first group is set-valued (with respect to one repair ID), the second group is scalar-valued. Both groups can be stored together without to introduce redundancies, for example by repeating identical values of scalar-valued attributes.

2.2 The Spreadsheet Import panel

2.2.1 Importing a simple, tabular spreadsheet

A flat tabular collection of data residing in a MS Excel spreadsheet can be imported in Synop Analyzer as a data source as follows:

- Select **File → Import data from spreadsheet** from Synop Analyzer's main menu.
- In the file selection dialog which opens up, choose the name of the Excel file which is to be opened.
- A new window named **Spreadsheet** opens up. In this window, you have to specify the name of the worksheet which contains your data (selector box **Sheet name**). The spreadsheet's first worksheet is preselected.
- Then you just have to press **Start transformation**. The data are read, and Synop Analyzer opens up an Input Data panel in which further user-defined data preparation and data specification steps can be performed.

2.2.2 Importing spreadsheets with a complex cell structure

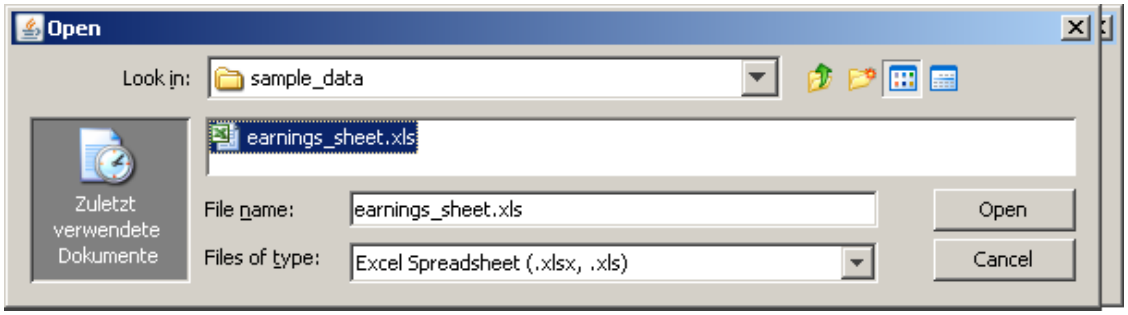
In the following sections we will explain the features and functions of the spreadsheet import wizard at the example of the MS Excel file `doc/sample_data/earnings_sheet.xls`. The file contains the monthly earnings sheet for a small company with two locations for the period from January 2006 to March 2009. The figure below shoes a part of this Excel sheet:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**Earnings Sheet 2006 - 2009**															
2		Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 2	Location 2
3		*01.2006*	*02.2006*	*03.2006*	*04.2006*	*05.2006*	*06.2006*	*07.2006*	*08.2006*	*09.2006*	*10,2006*	*11,2006*	*12,2006*	*13,2006*	*01.2006*	*02.2006*
5	**Total Sales**	**1403.6**	**1536.2**	**1117.8**	**981.0**	**874.5**	**1149.3**	**1288.2**	**1365,1**	**1070,9**	**923,0**	**845,8**	**881,3**		**863,7**	**851,9**
7	Subcontracting Cost	151,7	111,4	76,4	52,0	50,4	48,3	67,9	66,9	67,9	24,4	27,6	63,1	30,1	118,1	104,9
8	**Gross Profit I**	**1251,9**	**1424,8**	**1041,4**	**929,0**	**824,1**	**1101,0**	**1220,3**	**1298,2**	**1003,0**	**898,6**	**818,2**	**818,2**	**-30,1**	**745,6**	**747,0**
9	Personnel Cost	710,8	688,0	510,5	569,3	416,0	432,6	440,1	495,1	524,1	392,1	358,5	392,4		383,2	374,9
10	Supplies,Energy	144,4	108,4	85,3	66,9	35,3	32,0	45,0	35,1	60,9	51,7	50,9	117,3	12,5	70,8	23,6
11	Rental & Lease	31,3	27,6	29,8	24,4	153,2	155,6	155,6	150,6	153,1	151,4	153,2	154,9		73,8	73,8
12	Maintenance & Repair	22,2	15,7	15,6	12,9	8,7	10,5	33,3	38,1	17,9	12,5	7,4	29,1		1,2	6,9
13	Insurance,Fee,Taxes	15,2	13,4	14,7	13,4	13,4	13,4	13,4	14,2	13,4	13,4	13,4	13,4	39,3	4,2	4,2
14	Other Operating Charges	7,9	180,2	158,4	61,1	55,2	43,3	43,7	43,3	42,3	43,0	42,8	57,4	-44,2	4,0	3,5
15	Depreciations of Fixed Asse	177,3	177,3	177,3	177,3	117,1	118,8	118,8	118,8	118,8	118,8	118,8	133,8		43,5	43,5
16	**Gross Profit II**	**142,8**	**214,2**	**49,8**	**3,7**	**25,2**	**294,8**	**370,4**	**403,0**	**72,5**	**115,7**	**73,2**	**-80,1**	**-37,7**	**164,9**	**216,6**
17	Total Indirect Cost	78,8	88,7	48,7	61,9	53,0	41,2	47,7	37,3	45,2	45,3	47,5	15,4	-1,9	44,0	43,3
18	**EBIT**	**64,0**	**125,5**	**1,1**	**-58,2**	**-27,8**	**253,6**	**322,7**	**365,7**	**27,3**	**70,4**	**25,7**	**-95,5**	**-35,8**	**120,9**	**173,3**
19	Financial Income(Charges)	44,3	41,8	49,2	48,5	10,9	8,4	10,5	10,9	12,5	12,5	13,4	13,4	50,2	8,4	11,9
21	**EBT**	**19,7**	**83,7**	**-48,1**	**-106,7**	**-38,7**	**245,2**	**312,2**	**354,8**	**14,8**	**57,9**	**12,3**	**-108,9**	**-86,0**	**112,5**	**161,4**

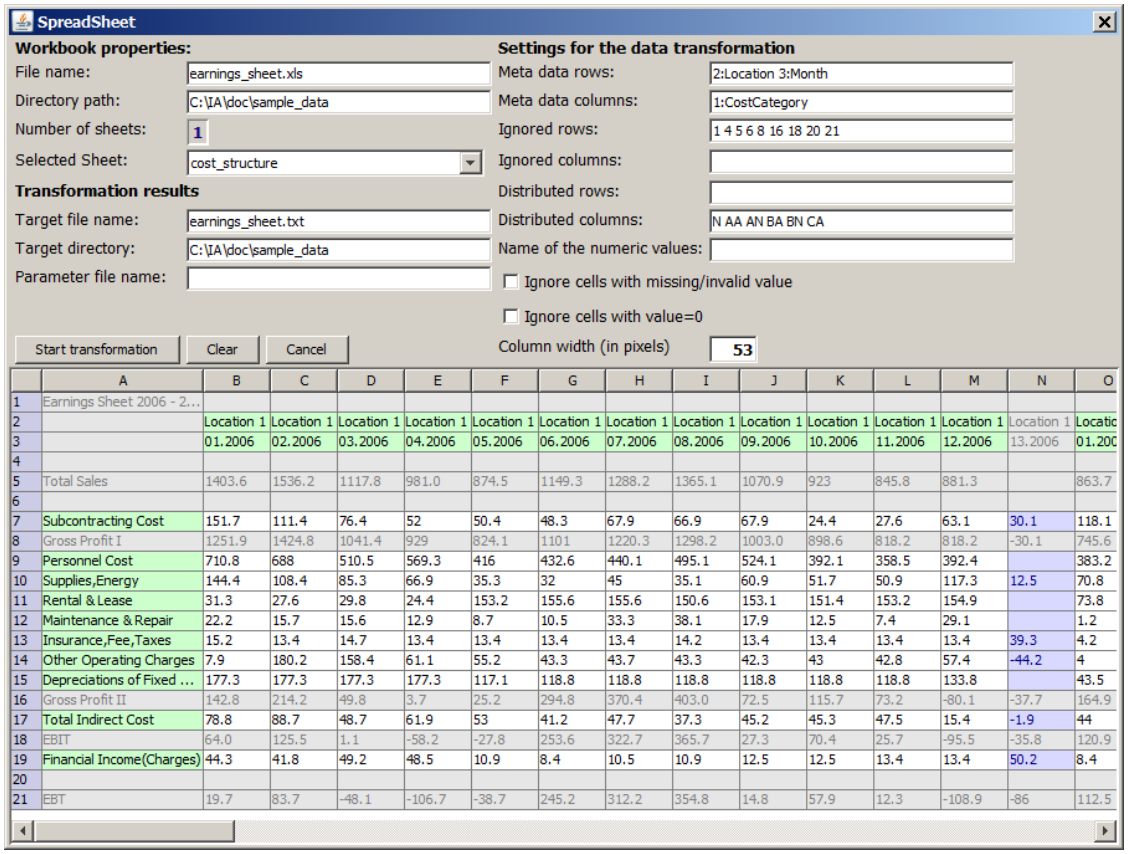
In the present form, the data are not really suitable for being used by a forecasting and trend analysis tool: in the Excel sheet, meta data information (such as location, date or cost category) is intermixed with number cells, empty space cells, formula cells (such as EBT or Gross Profit II) and auxiliary title or text cells. Furthermore, the sheet contains accountant's corrections at year end (such as the column `13.2008` highlighted in the picture above); these corrections have to be distributed on the 12 months of the preceding year before the corresponding time series can be used for a forecast or trend

analysis. We will see that Synop Analyzer supports various preprocessing steps on this input sheet in order to overcome the aforementioned problems.

From the Synop Analyzer main menu, we select **File → Import data from spreadsheet**. A file chooser dialog opens up.



We select the file `doc/sample_data/earnings_sheet.xls` in the file chooser dialog. A new **Spreadsheet** window opens on the main canvas:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Earnings Sheet 2006 - 2...														
2		Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Locatic
3		01.2006	02.2006	03.2006	04.2006	05.2006	06.2006	07.2006	08.2006	09.2006	10.2006	11.2006	12.2006	13.2006	01.200
4															
5	Total Sales	1403.6	1536.2	1117.8	981.0	874.5	1149.3	1288.2	1365.1	1070.9	923	845.8	881.3		863.7
6															
7	Subcontracting Cost	151.7	111.4	76.4	52	50.4	48.3	67.9	66.9	67.9	24.4	27.6	63.1	30.1	118.1
8	Gross Profit I	1251.9	1424.8	1041.4	929	824.1	1101	1220.3	1298.2	1003.0	898.6	818.2	818.2	-30.1	745.6
9	Personnel Cost	710.8	688	510.5	569.3	416	432.6	440.1	495.1	524.1	392.1	358.5	392.4		383.2
10	Supplies,Energy	144.4	108.4	85.3	66.9	35.3	32	45	35.1	60.9	51.7	50.9	117.3	12.5	70.8
11	Rental & Lease	31.3	27.6	29.8	24.4	153.2	155.6	155.6	150.6	153.1	151.4	153.2	154.9		73.8
12	Maintenance & Repair	22.2	15.7	15.6	12.9	8.7	10.5	33.3	38.1	17.9	12.5	7.4	29.1		1.2
13	Insurance,Fee,Taxes	15.2	13.4	14.7	13.4	13.4	13.4	13.4	14.2	13.4	13.4	13.4	13.4	39.3	4.2
14	Other Operating Charges	7.9	180.2	158.4	61.1	55.2	43.3	43.7	43.3	42.3	43	42.8	57.4	-44.2	4
15	Depreciations of Fixed ...	177.3	177.3	177.3	177.3	117.1	118.8	118.8	118.8	118.8	118.8	118.8	133.8		43.5
16	Gross Profit II	142.8	214.2	49.8	3.7	25.2	294.8	370.4	403.0	72.5	115.7	73.2	-80.1	-37.7	164.9
17	Total Indirect Cost	78.8	88.7	48.7	61.9	53	41.2	47.7	37.3	45.2	45.3	47.5	15.4	-1.9	44
18	EBIT	64.0	125.5	1.1	-58.2	-27.8	253.6	322.7	365.7	27.3	70.4	25.7	-95.5	-35.8	120.9
19	Financial Income(Charges)	44.3	41.8	49.2	48.5	10.9	8.4	10.5	10.9	12.5	12.5	13.4	13.4	50.2	8.4
20															
21	EBT	19.7	83.7	-48.1	-106.7	-38.7	245.2	312.2	354.8	14.8	57.9	12.3	-108.9	-86	112.5

The upper right part of the window contains several input fields in which we can specify how the spreadsheet is to be used. The lower part of the window shows the effects of these specifications.

- In the field **Meta data rows** we specify that the second and third row of the Excel sheet contain two different kinds of meta data information which we would like to use in our analysis. By typing `2:Location 3:Month` we indicate that we want to refer to the meta data in row 2 under the label `Location` and to the meta data in row 3 under the label `Month`.
- Similarly, we indicate that the first column (`A`) of the Excel sheet contains a meta data information to which we want to refer under the label `CostCategory`.
- Our goal in this example is a cost structure analysis. Therefore, we only maintain the rows containing the various cost category figures and we discard the other figures such as `Total Sales`, `Gross Profit`, `EBIT` or `EBT`. That's why we type `1 4 5 6 8 16 18 20 21` into the field **Ignored rows**.
- The columns `N`, `AA`, `AN`, `BA`, `BN` and `CA` of the Excel sheet contain the accountant's corrections at year-end for the two locations. We want to distribute these corrections equally on the 12 months preceding the correction and discard the correction month 13. Therefore, we enter `N AA AN BA BN CA` in the input field **Distributed columns**.

The specifications decribed above are automatically reflected by an adapted coloring scheme in the tabular representation of the currently active spreadsheet in the lower part of the screen: spreadsheet cells containing meta data information are displayed with a green background, cells with values to be distributed among other cells have a blue background, cells which are to be ignored are grayed out and 'normal' value cells are displayed with white background.

Finally, we click on the **Start Transformation** button. An instant later, the pop-up window closes and the transformed flat file `earnings_sheet.txt` is written into our chosen target directory.

The generated file contains the columns `Location`, `Month`, `CostCategory` and `Cost`. The new file is suitable for a statistical analysis using the entire set of Synop Analyzer functions. The new file is suitable for being read into Synop Analyzer, and it is automatically opened in the Input Data panel.

2.2.3 Reusing spreadsheet import tasks

If you enter a file name into the input field named **Parameter file name** in the spreadsheet import opp-up window (the file name should end with `.xml`), then the specifications that you perform in the pop-up window will be saved to that file automatically when you press the button **Start transformation**. You can later load these settings by selecting **File → Import data from spreadsheet**, by changing the file type to **parameter file (.xml)** in the file chooser dialog and by navigating to the previously stored parameter file.

You can also save and re-load spreadsheet import settings as a part of a larger data import process. To that purpose, leave the spreadsheet import window by pressing **Start transformation**, optionally perform further data import settings in the left column of

the main window and save the settings by selecting **File → Save Data Load Task**. In this case, your spreadsheet import settings are stored within the resulting parameter file. You can later execute this data load task by selecting **File → Open Data Load Task**.

2.3 The Google Analytics Data Import module

2.3.1 Google Analytics

Google Analytics (see http://www.google.com/analytics/) is a mechanism for tracking the usage statistics and typical browsing paths of web sites and web shops offered by Google Inc.. The basic service is free, only high-volume usage is charged.

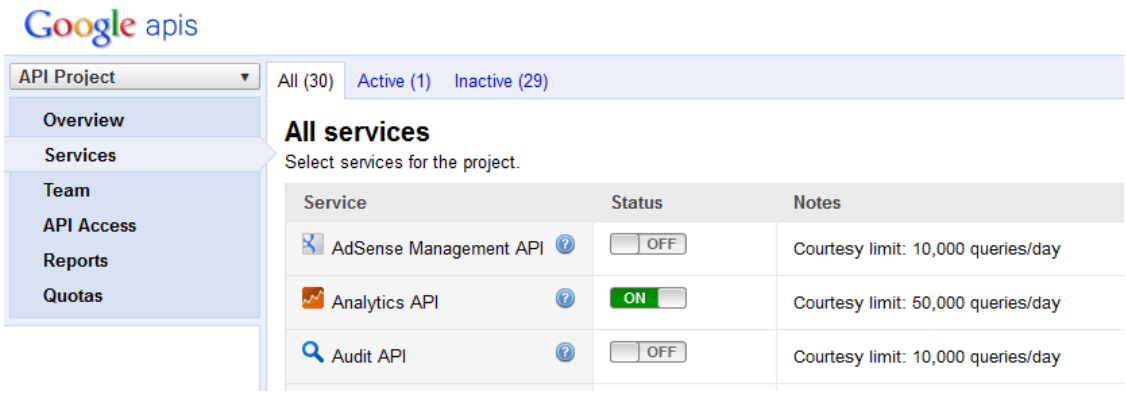
Two actions must be taken in order to use Google Analytics for a web site. First, on http://www.google.com/analytics/ an account must be created in which the domains and sub-domains to be tracked are specified. Second, each web page whose usage is to be tracked must be equipped with a little script which sends tracking information to the Google Analytics database each time the web page is opened. An explanation and step-by-step instructions can be found at http://www.google.com/analytics/discover_-analytics.html.

Within a Google Analytics account one can define one or more 'web properties' and within each web property one or more 'profiles'. A web property can be regarded as a group of interrelated analytics tasks and each profile as one single analytics task. By inserting the little tracking scripts into the single web pages one defines which web page will send usage information to which profile.

For Evaluating the collected results, Google Analytics provides both a browser-based graphical frontend and an application programming interface (API) via which the collected data of a profile can be read into a third party program. This API is used by Synop Analyzer for reading the Google Analytics data of a web site into the software and for interactively exploring them.

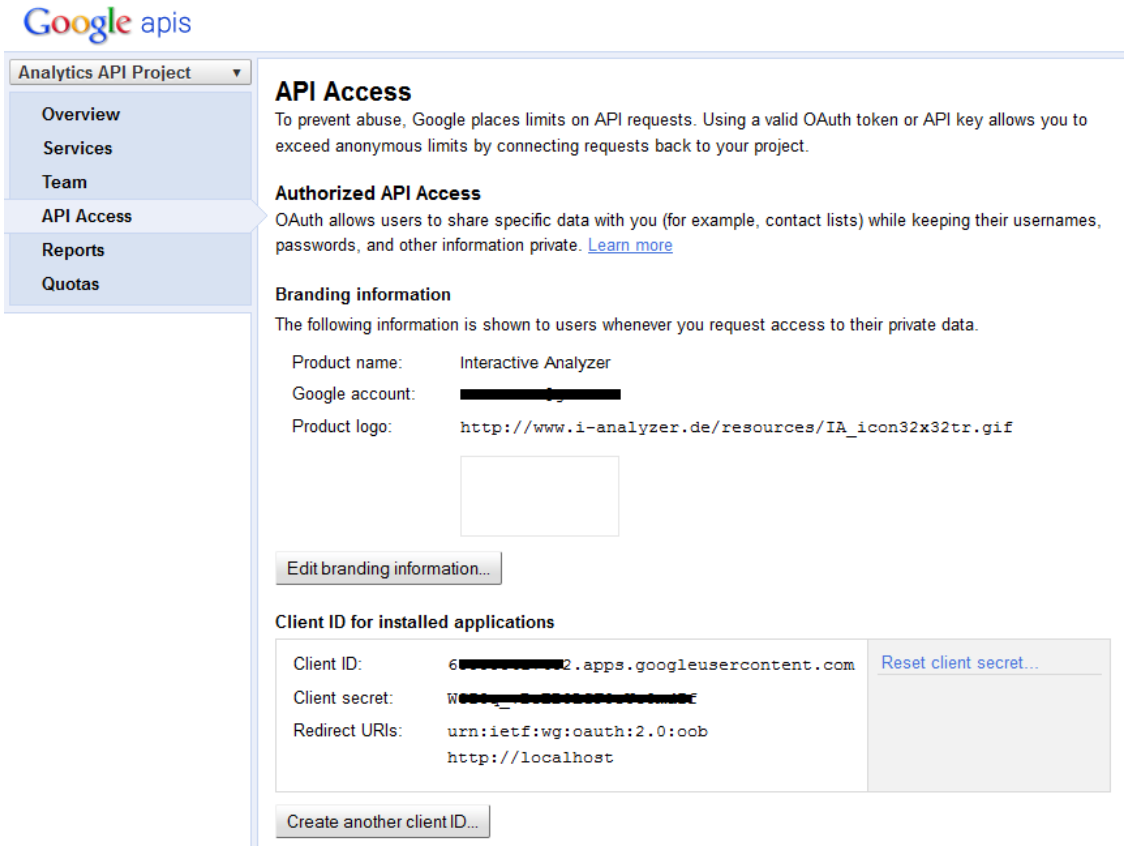
2.3.2 Reading data via the Google Analytics Reporting API

In order to be able to access a web site's collected Google Analytics data, the owner of the Google Analytics account has to make sure the Analytics API service is enabled. To that purpose, log into the API administration console https://code.google.com/apis/console/, click on the menu item **Services** and switch the status of the service **Analytics API** to 'ON'.



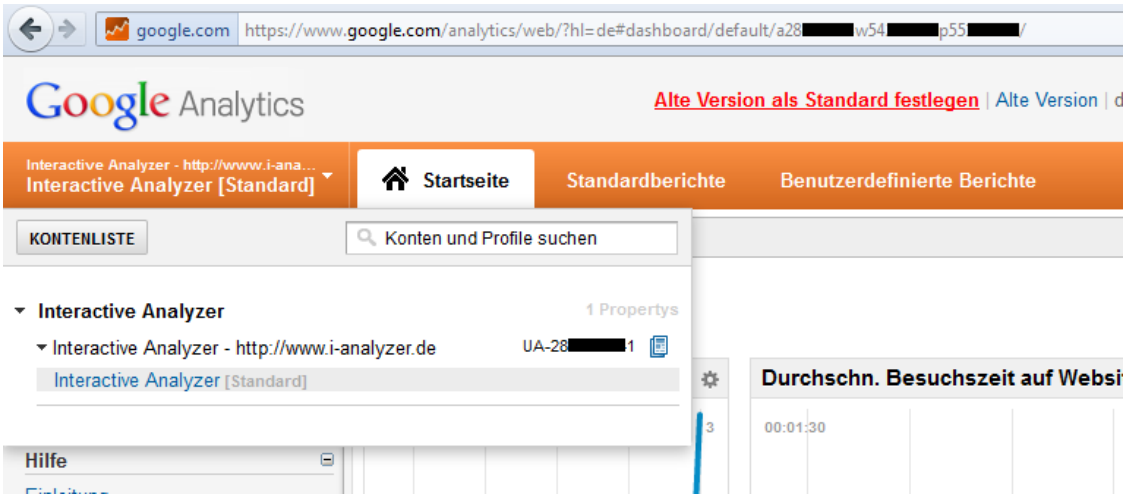
Next, you have to create at least one access client within the API administration console. A client is a predefined access path for external programs which grants access to the collected data of one or more profiles within a Google analytics account.

To create a new access client, click on the menu item **API Access** within this administration console. This opens up a screen view in which pressing the button **Create Client ID** creates a new access client consisting of a client ID, a password called 'client secret' and a 'redirect URL'.


Google apis
Analytics API Project
Overview
Services
Team
API Access
Reports
Quotas
API Access
To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.
Authorized API Access
OAuth allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. Learn more
Branding information
The following information is shown to users whenever you request access to their private data.
Product name: Interactive Analyzer
Google account:
Product logo: http://www.i-analyzer.de/resources/IA_icon32x32tr.gif
Edit branding information...
Client ID for installed applications
Client ID:
.apps.googleusercontent.com
Reset client secret...
Client secret:
Redirect URIs: urn:ietf:wg:oauth:2.0:oob
http://localhost
Create another client ID...

When you later try to connect to the Google Analytics API, you must have these three values (client ID, client secret and redirect URL) at hand.

In addition, you must specify the 8-digit ID of one of the predefined profiles within the Google Analytics account. You can step through all available profile IDs by logging into your accoutn at https://www.google.com/analytics/web/ and by then clicking on the menu item on the upper left corner of the screen. This menu item opens a drop-down list of all profiles within the active Google Analytics account. A mouse-click on one of the rows of this drow-down list makes the account ID, web property ID and profile ID of that entry appear in the URL address line in your web browser. The profile ID is the series of 8 digits at the very end of the displayed URL, right after the letter 'p'.



These 8 digits (in the picture below they start with `55...`) must also be at hand when you try to read Google Analytics data from Synop Analyzer.

2.3.3 The panel for specifying a Google Analytics data source

From Synop Analyzer, you access a Google Analytics data source by clicking **File → Import data from Google Analytics API**. This action opens up a bew panel in which the previously described values of client ID, client secret and profile ID can be entered.

Import Google Analytics data
Client ID for Google Analytics API
Client Secret for Google Analytics API
Profile ID for Google Analytics API
Dimensions to be read from Google Analytics API
ga:source,ga:medium
Metrics to be read from Google Analytics API
ga:visits,ga:pageviews
Redirect URL for Google Analytics API
urn:ietf:wg:oauth:2.0:oob
Scope for Google Analytics API
https://www.googleapis.com/auth/analytics.readonly
Start date for the extracted data
2012-01-01
Authorization code provided by Google Analytics API
Generate authorization code
Read and store data
Cancel

In the panel fields **Dimensions to be read from Google Analytics API** and **Metrics to be read from Google Analytics API** you specify which types of information, that means which data fields the retrieved data table will have. Each single dimension and each metrics has to be entered in the form of `ga:[Name]`. A list of all supported dimension and metrics name can be found at http://code.google.com/intl/de-DE/apis/analytics/docs/-gdata/dimsmets/dimsmets.html. The specifications performed in the screenshot above specify that the retrieved data has the four columns `ga:source` (web domain, from which the visitor came to our tracked web site), `ga:medium` (type of the web site from which the visitor came to our web site), `ga:visits` (number of visits to our tracked web site) and `ga:pageviews` (number of clicked web pages).

You can memorize frequently used client IDs, client secrets, profile IDs, active dimensions and active metrics in the preference settings at <buf>Preferences → Data Import Preferences</buf>. Then you do not have to type in these values manually each time you open the Google API sepecification panel.

And one more step has to be done befor the data transfer can succeede: an authorization code must be created and entered into the last input field of Synop Analyzer's Google Analytics API specification panel. You generate an authorization code by pressing the button **Read and Store data** in the panel, by accepting the security question in the browser window which pops up and by copying the displayed code into the Synop Analyzer panel.

Once all input fields of the panel have been filled correcty, pressing the button **Read and store data** the data retrieval process. The data are first saved to a local file named `ga[ProfileID]_[currentDa te].txt` in the crrent working directory; afterwards, they are read into a data source tab on the left side of the Synop Analyzer workbench just as any other input data source.

Note that the authorization code is not reusable: you have to create and fill in a new authorization code each time you want to read data from the API.

2.4 Data Transformations

2.4.1 Purpose

The 'Data Transformation' functions in the data source panel can be used to transform an existing in-memory data source within Synop Analyzer into one or two new data sources with slightly different properties. The new data sources will be available in Synop Analyzer in addition to the original data source.

At the moment, the following data transformation functions are available:

- :Group data rows: the data records of the original data source are grouped (aggregated) into larger groups. A numeric data field serves as the grouping criterion: a new group begins whenever the value of this data field differs from the previous record's value or the value on the first record of the group by more than a user-defined threshold.

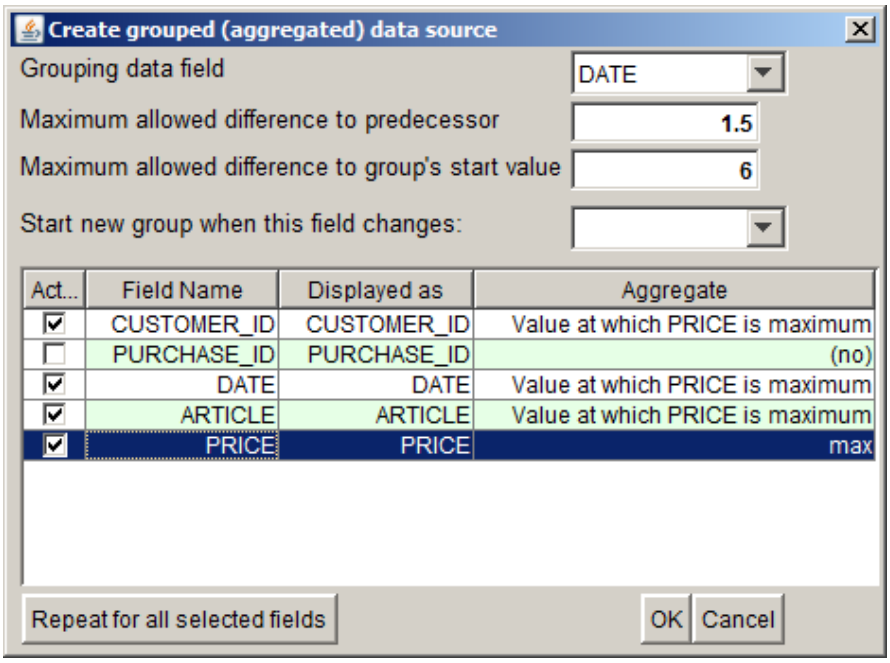
- :Split the data: the original data source is split into two parts. Each data record of the original data is assigned to exactly one of the two new parts. The assignment is performed by means of a random number generator. The data can be split symmetrically (50:50) or asymmetrically.

2.4.2 Aggregating (grouping) data records

This transformation function creates a new data source in which the data records are aggregated into larger groups, each group defining one data record of the new data. Optionally, some of the data fields of the original data source can be suppressed during that transformation.

In the following paragraphs, we will demonstrate that function using a concrete example. To that purpose, we open the data `doc/sample_data/RETAIL_PURCHASES_BY_TIME.txt` and read them into Synop Analyzer using the default settings. The file contains supermarket checkout data: 1000 purchased articles, sorted by the date and time of purchase.

We want to create a list of the most expensive purchase article (and the customer who purchased it) of each week. By clicking the button , we open a pop-up window in which we can specify the parameters for a data aggregation:



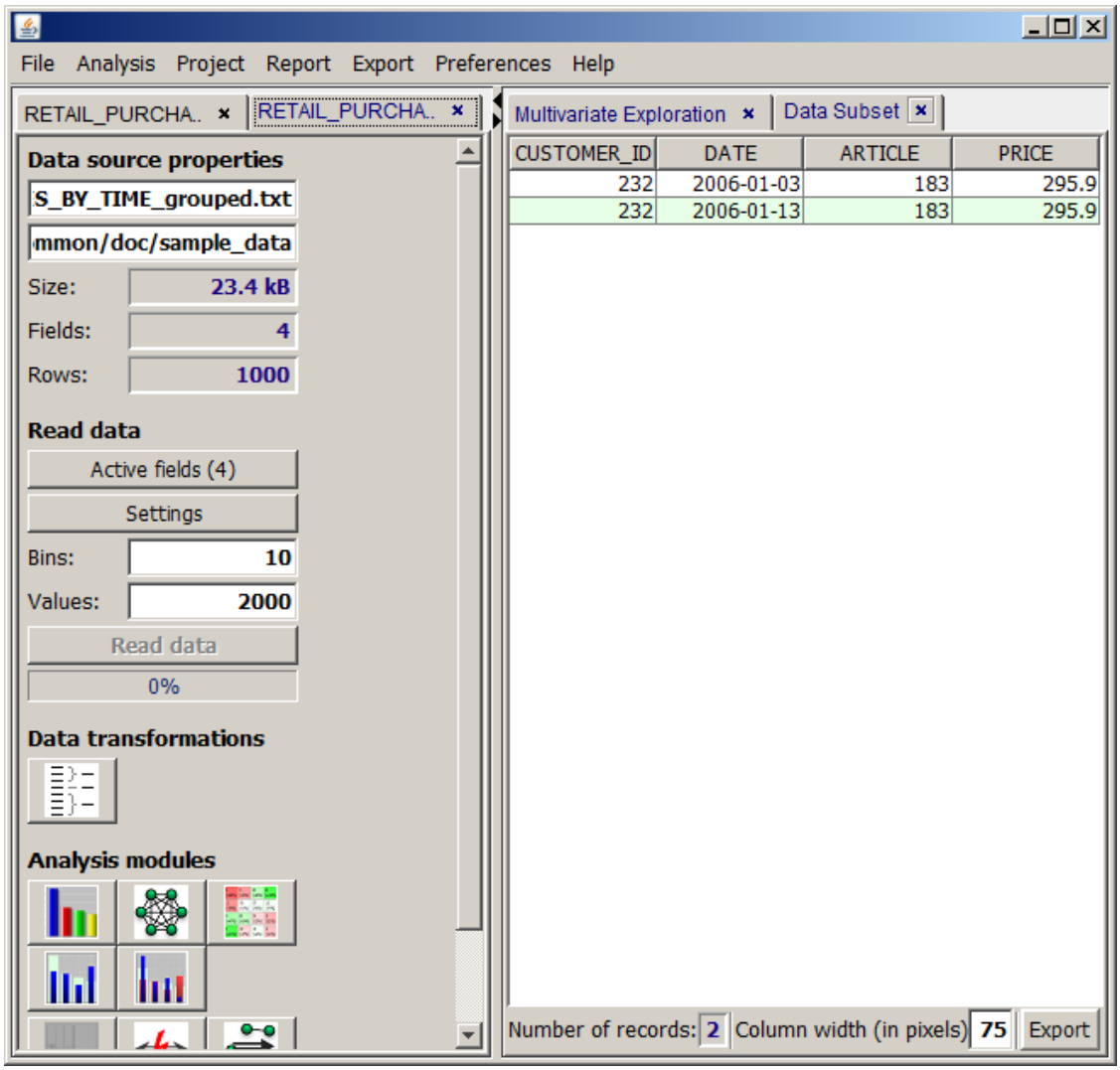
In the screenshot shown abive, we have already performed the following modifications in the panel:

- In ths selection field **Grouping data field**, we have selected the data field `DATE`. This data field and its values will serve as the grouping criterion for the aggregation; it will help us to group the data by weeks.
- In the input field **Maximum allowed difference to predecessor**, we can specify one of the criteria which define where one group ends and the next group begins. We are interested in groups starting on Monday morning and ending on Saturday evening. Therefore, we enter the value '1.5'. That means, we want a group to be terminated when a period of 1.5 days is found without any transaction (Sunday).
- In the input field **Maximum allowed difference to group's start value** we can specify an additional group separator criterion. This one compares the current record's value of the grouping field to the corresponding value on the first data record of the current group; it terminates the group and starts a new one if the value difference exceeds a threshold. We enter '6' here, thereby specifying that a group should end 6 days after the first `DATE` value of the group. In our case, this criterion is redundant to the criterion specified in the line above, we could have left that field empty.
- In the selection field **Start new group when this field changes**, we can enter an additional 'hard' group seperator criterion. If we selected the field `CUSTOMER_-ID</but> here, a new group would be started each time the value of the field <data>CUSTOMER_ID` differs from this field's value on the previous data record.
- The table in the center of the pop-up window lists all data fields which are available in the original data source. In the column **Active** we can suppress certain data fields,

in the column **Displayed as** we can assign new display names to certain data fields. The value selected in column **Aggregate** specifies how the different values of the corresponding data field on the different data records of a newly formed data group are agregated in order to get the group's value of that data field. The default setting is that numeric data fields are summed up, for date/time fields, the mean date or time on the group is calculated and for textual data fields, all different values are separately kept, making the field a set-valued field in the aggregated data.
In our example, we have suppressed the field `PURCHASE_ID`. In the field `PRICE` we want to get the price of the most expensive article within the group; in the fields `CUSTOMER_ID`, `DATE` and `ARTICLE` we want to see the customer ID, the purchase date and the article ID of the most expensive purchased article within the group.

- The button **Repeat for all selected fields** serves to repeat an action which has been performed on one single data field on all currently selected (blue) rows of the table. This function is helpful on data sources with a large number of data fields.

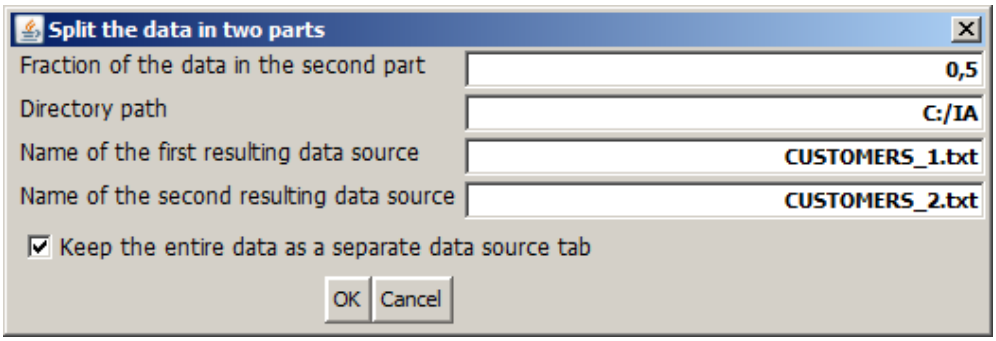
By clicking the **OK** button we start the data transformation. A new tab pops up on the left side of the Synop Analyzer workbench. The new tab contains the transformed data source and offers the same functional buttons for data transformations and data analysis functions as the input data tab of the original data source. Clicking the button and then in the new window the button shows the data records of the new, aggregated data source. As expected, the new data contain only two records, one for each week covered by the original data. Surprisingly, on each of the two weeks, the most expensive purchased article was the same one and it was purchased by the same customer.



2.4.3 Splitting a data source in two parts

This transformation splits the data in two parts. Each data record of the original data is assigned to exactly one of the two new parts. The assignment is performed by means of a random number generator. The data can be split symmetrically (50:50) or asymmetrically.

Clicking the button opens the following pop-up dialog:



In the first input field of the dialog, we define the size ratio of the two data parts. The predefined value of 0.5 creates two parts of equal size.

The second, third and fourth Input field specify the directory path, the names and - indirectly via the file name endings - the types of the files in which the resulting partial data are to be persistently stored on disk. Leave these fields empty if you do not want to store the data parts persistently. Hence, the following alternatives are possible:

- No entry in the data name input field: the data are not stored on disk. You do not have the possibility to re-read the partial data with modified settings during the current Synop Analyzer session. If the current analyses are stored as a project and the project is reopened later, each data part must be recreated by reading the entire data and sorting out a part of it. That can consume much time on large data. On the other hand, that variant avoids the risk of unwillingly working on outdated partial data once the original data source is updated.
- An entry with ending `.iad` in the file name field: the data are stored on disk in the proprietary compressed iad format. That means there is no possibility to re-read the data parts with modified settings. But if the current analysis project is stored and reopened later, the data parts will be imported very fast, even if the data are large.
- An entry with ending `.txt` in the file name field: the data are stored on disk as flat text files. That means there is the possibility to re-read the data parts with modified settings during the current Synop Analyzer session. If the current analysis project is stored and reopened later, the data parts will be imported from the two flat files, which is faster than reading the entire original data twice, but much slower than importing two iad files.

Finally, the check box specifies whether the original data is maintained as a separate input data tab within %IA;, or whether the original data tab is replaced by the first resulting part after the split.

3

Data Analysis and Visualization Modules

This part of the user's guide contains a reference documentation of all data analysis and data visualization modules of Synop Analyzer. Depending on your license, not all of the modules described here may be visible for you. You can activate those modules by updating your license.

Statistics and Distributions: The module 'Statistics and Distributions' presents an overview over all available data attributes, their statistical properties and their value distributions.

Correlations Analysis: The Correlations Analysis panel computes and displays field-field correlations between the available data attributes (fields); it provides the 'drill-down' into a single pair of data fields by means of Bivariate Exploration.

Bivariate Exploration: The Bivariate Exploration panel provides a refinement of the field-field **Correlations Analysis**: for a given pair of data fields, it presents a matrix of value-value interrelations and offers further interactive drill-down capabilities.

Pivot Tables: The Pivot Tables panel creates aggregation tables which show the values of a user-defined statistical measure of the data as a function of the value ranges of two or more data fields.

Multivariate Exploration: The Multivariate Exploration panel provides interactive multi-dimensional ad-hoc analysis and drill down features with real-time response even on multi gigabyte data.

Split Analysis: In the Split Analysis panel, two disjunct data subsets can be defined: test data and control data. The control data can be further sampled in order to become representative for the test data with respect to certain data fields. On the other data fields, significant deviations between the test and the control data can be studied and quantified.

Time Series Analysis: In the Time Series Analysis panel, trends and seasonal patterns in time series data can be detected, and future values can be forecasted.

Deviations, Inconsistencies: In the Deviation Detection module, outliers, deviations and presumable data inconsistencies can be detected. The specific approach of this module is that it does not examine the values and value distribution characteristics of each data field separately for outliers as traditional data quality checker tools do. Rather, it finds cross-field inconsistencies.

Associations Analysis: An Associations Analysis detects typical patterns or atypical deviations in the data.

Sequences Analysis: Sequences Analysis, also called Sequential Pattern Analysis, is a refinement of **Associations Analysis**: it detects time-ordered patterns and is a means for detecting causal relations in the data.

Self Organizing Maps (SOM): Self Organizing Maps (SOM) is a neural network approach in which a two-dimensional net of neurons 'learns' the training data. Afterwards, the SOM net can be used to detect homogeneous clusters in both the training data and new data sources, or for predicting missing values within these data.

Linear und Logistische Regression: Linear and Logistic Regression are basic Data Mining techniques which try to predict the values of one data field, the target field, using the values of other data fields and grouping them into a linear equation. Linear regression is suitable for numeric target fields, logistic regression for two-valued data fields with values such as male/female, yes/no or 0/1.

3.1 The Module 'Statistics and Distributions'

3.1.1 Purpose and short description

The data exploration module 'Statistics and Distributions' is the easiest, most fundamental data visualization module of Synop Analyzer. The screen screen is vertically divided into two areas. The upper part contains some basic statistical measures and figures of each data field in tabluar form. The lower part shows the value distribution of each data field in the form of histogram charts. In summary, the purpose of the module is to give a quick overview over a data source which has been read into Synop Analyzer:

- Which attributes (data fields) are available in the data, which data type do they have and which values do they contain?
- How well are the data fields filled? Where are major gaps and many missing or invalid values?
- Are the available values reasonable? Are there obvious deviations?
- What are the most frequent values? What is the form of the value distribution curve for numeric data fields? Gaussian? Equally distributed? Logarithmic?
- Which automatically generated value ranges and interval boundaries should be manually modified in order to get maximally meaningful histogram charts?

3.1.2 The tabular views

In the upper part of the module 'Statistics and Distributions' two tabular views display important statistical measures of the numeric and the non-numeric data fields. The screenshot below shows these tabular views for the data `doc/sample_data/RETAIL_PURCHASES.txt`, which have been imported into Synop Analyzer as described in Importing data with name mappings:

Statistics and Distributions

Numeric/Group data field	Invalid or NULL	Diff. values	Minimum	Maximum	Second sm...	Second lar...	Mean	Median	Std. dev...	Skewness	Excess
PURCHASE_ID length	0	14	1.0	28.0	2.0	19.0	3.7313433	2.5	3.4612226	3.9377024	20.02698
DATE	0	12	2006-01-02	2006-01-14	2006-01-03	2006-01-14	2006-01-08	2006-01-09	4.4409385	-0.08885241	-1.5540137
PRICE	0	127	1.18	744.75	1.29	338.76	41.700745	7.5	81.92312	5.7247877	40.767857

Textual data field	Invalid or NULL	Diff. values	Other values	Most freq. value	Highest freq.	Second most freq. value	2nd highest freq.
ARTICLE	0	79	0	lemonade	50	cream	47

The screenshot shows that the textual data field `ARTICLE` has no missing values, that it has 79 different values and that the value `lemonade`, which occurs in 50 purchase IDs, is the most frequently purchased article, followed by the article `cream` contained in 47 purchases.

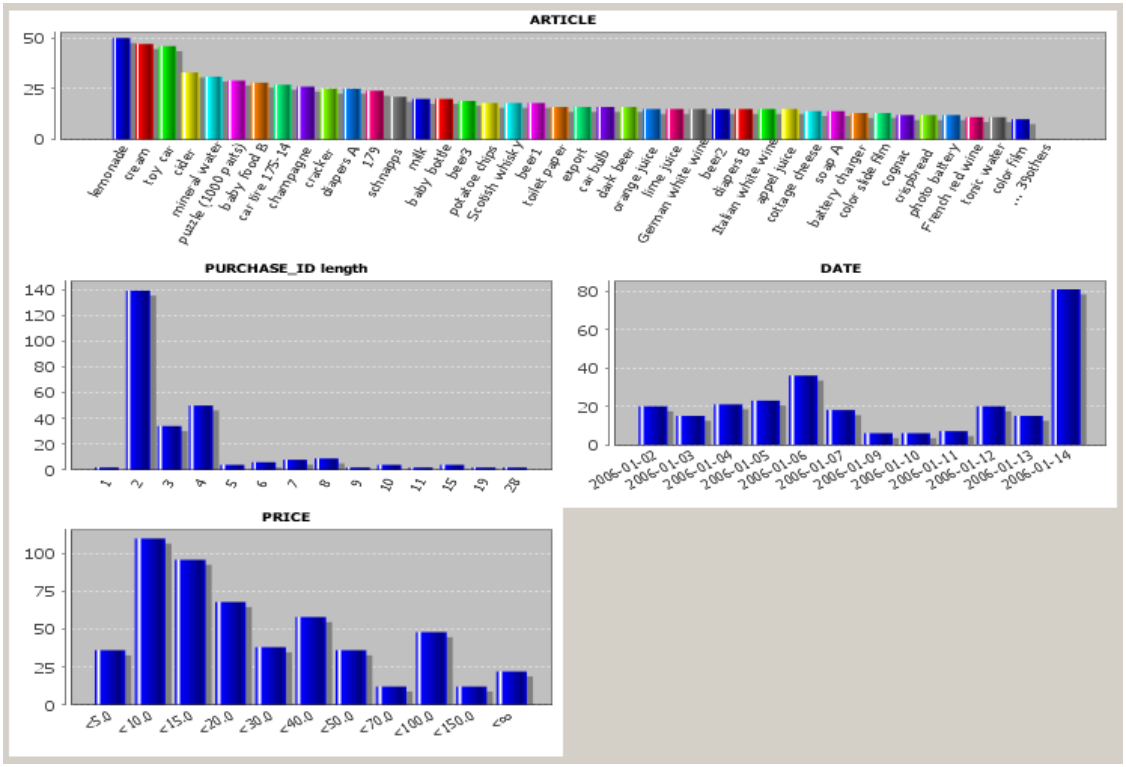
For the numeric field `PRICE` we see that it has no missing/invalid values either, that the cheapest purchase was 1.18 , *themostexpensiveone*744.75, the average purchase value was

41.70 $but 50\% of all purchases were below 7.50$. That means, there are many small purchases and a few very large ones. Accordingly, the distribution of purchase prices has a positive skewness (long tail towards high prices). A precise definition of the three measures 'Standard deviation', 'Skewness' and 'Excess' can be found on the following Wikipedia pages: Sample standard deviation, Skewness und Excess Kurtosis.

For group fields (in our example the field `PURCHASE_ID`, Synop Analyzer does not show a statistics on the field values - the field values of group fields are normally of little interest since they are only used to define groups of data records. Instead, a statistics and distribution of group lengths is shown, in other words: a statistics on how many data records are in the various data groups defined by identical group field values.

3.1.3 The histogram charts view

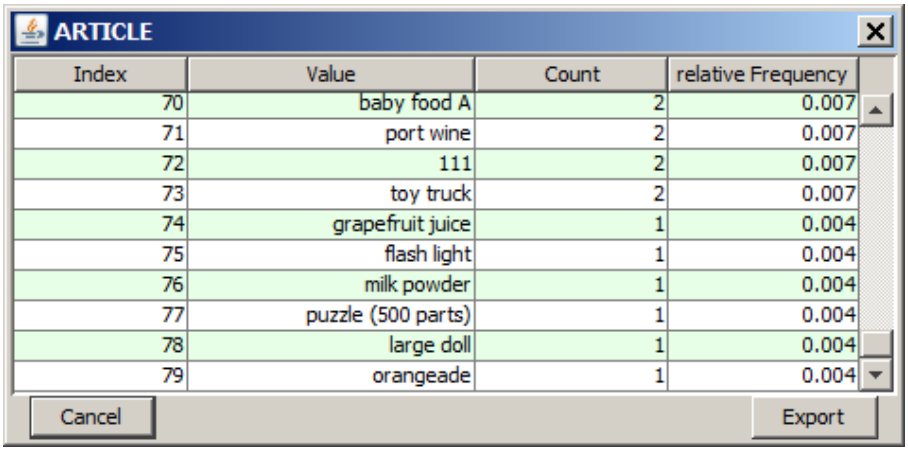
The lower part of the screen shows value distribution histograms for all data fields. Histograms with more than 40 bars cover the entire screen width, histograms with not more than 20 bars are grouped into tupels of N charts per screen row, where N is the number entered into the tool bar input field named **Charts/row**. If this input field contains the value 0, the software decides autonomously how many charts to put into one screen row. Charts with 21 to 40 bars occupy twice as much horizontal space as the charts with not more than 20 bars. In the figure below, we show value distribution histograms which have been generated on the sample data `doc/sample_data/RETAIL_PURCHASES.txt` after importig them as described in Importing data with name mappings.



In the histogram charts for non-numeric data fields, the values are arranged by descending occurrence frequency from left to right. Each value has another bar color. If a data field has more then N values, where N is the number in the input field **#values (text fields)** in the Input Data panel, then only the N most frequent values have been separately recorded when the data were imported. All other values have been summarized into the 'rest' value 'others'. This rest value will be represented in the chart by one single bar with label 'others'. If there is no such 'rest' value in the data, it can still be the case that there are so many different values that it is impossible to draw a histogram bar for each of them. In this case, the histogram chart will be truncated after 80 bars (you can change that value of 80 in the pop-up dialog **Preferences → Univariate Preferences**). The fact that some bars could not be displayed is indicated by an additional label saying "... ?? others", where ?? is the number of suppressed bars. The chart for the field `ARTICLE` in the figure above has such a label saying "... 39 others".

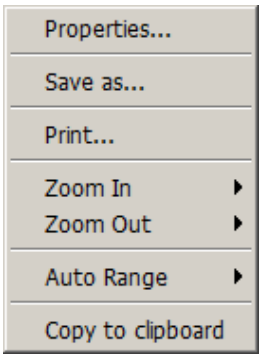
In the histogram charts for numeric data fields, all bars have the same color, and the values or value ranges are ordered by increasing value from left to right. A histogram for a numeric data field has - unless a manual field-specific discretization has been defined - never more than n histogram bars, where n is the number entered into the input field **#bins (numeric fields)** in the Input Data panel.

By left mouse click on a histogram chart you open a tabular detail view containing all different values of the field and their absolute and relative occurrence frequencies. This detail view also contains those values for which no separate bar could be drawn in the histogram charts due to lack of space. In the figure below, such a pop-up table view for the data field `ARTICLE` is shown.



By drawing with the mouse (keep the left mouse button pressed while moving) on a histogram chart you mark a rectangular region in which you want to zoom in.

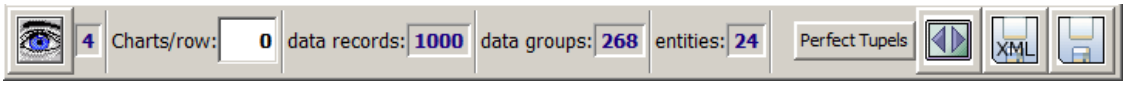
By right-clicking on a histogram chart you open the pop-up dialog shown below. In this dialog, you can modify the appearance of the histogram chart (text fonts and sizes, axis styles, labels, etc.) via the menu item **Properties**. You can also save the chart as PNG graphics, print it or copy it as png graphics object to the system clipboard.



Using the button **Visible fields** in the bottom toolbar, you can hide and remove certain fields from the charts panel in order to get a clearly arranged picture on data with many data fields.

3.1.4 The bottom tool bar

The tool bar at the lower screen border provides the following buttons and functions:



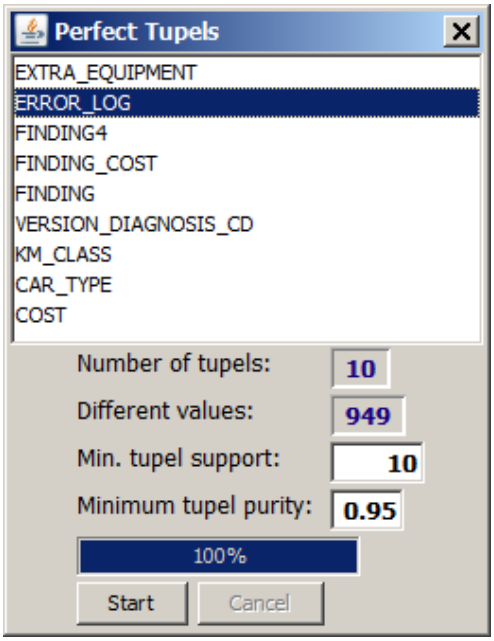
- :
  Via this button you open a pop-up dialog which permits to hide certain data fields from the histogram chart panel. The blue number to the right of the **Visible fields** button shows the total number of remaining visible fields.
- **Charts/row**
  In this input field you can specify how many of the 'normal' histogram charts with not more than 20 bars should be put into one single screen row. The smaller the number, the larger will be each single histogram chart.
- **Perfect tupels:**
  The purpose of this button will be described in section Detecting perfect tupels.
- :
  By clicking on this button you re-draw all histogram charts, thereby adapting their size to the current screen width.
- :
  By pressing this button, you can save the currently active data import settings and all settings performed in this module to a persistent XML parameter file. This file can later be opened via Synop Analyzer's main menu (**Analysis → Run Statistics and Distributions**). In this way you can exactly reproduce the current data analysis screen without to be obliged to re-enter all settings and customizations.

- :
  Export the current data exploration results within this module into a spreadsheet in `.xlsx` format (MS-Excel© 2007+). The spreadsheet contains several worksheets: one with a single png graphics for each histogram chart, one with a single png graphics for all charts, two for the two statistics tables, and one more worksheet for each detail pop-up window which ever has been opened by mouse-clicking on one of the histogram charts.

3.1.5 Detecting and removing perfect tupels

The detection of 'perfect tupels' is started by clicking on the tool bar button **Perfect tupels**. The button is only usable if a group field has been specified on the input data and if at least one of the textual input data fields is set-valued with respect to the group field, that means it contains more than one different value on at least some of the data groups. The button opens a pop-up dialog in which you can choose one of the set-valued fields and then search this field for 'perfect tupels'. A 'perfect tupel' is a set of two or more field values which occur always or almost always together in the same data groups.

In the following we want to demonstrate this using the sample data `doc/sample_data/-CAR_REPAIR.txt`. We assume that these data have been imported into Synop Analyzer as described in Transaktional data and stream data, that means with `REPAIR_ID` as group field. If we start the module 'Statistics and Distributions' on these data and click on the **Perfect tupels** button, the following dialog window opens up:

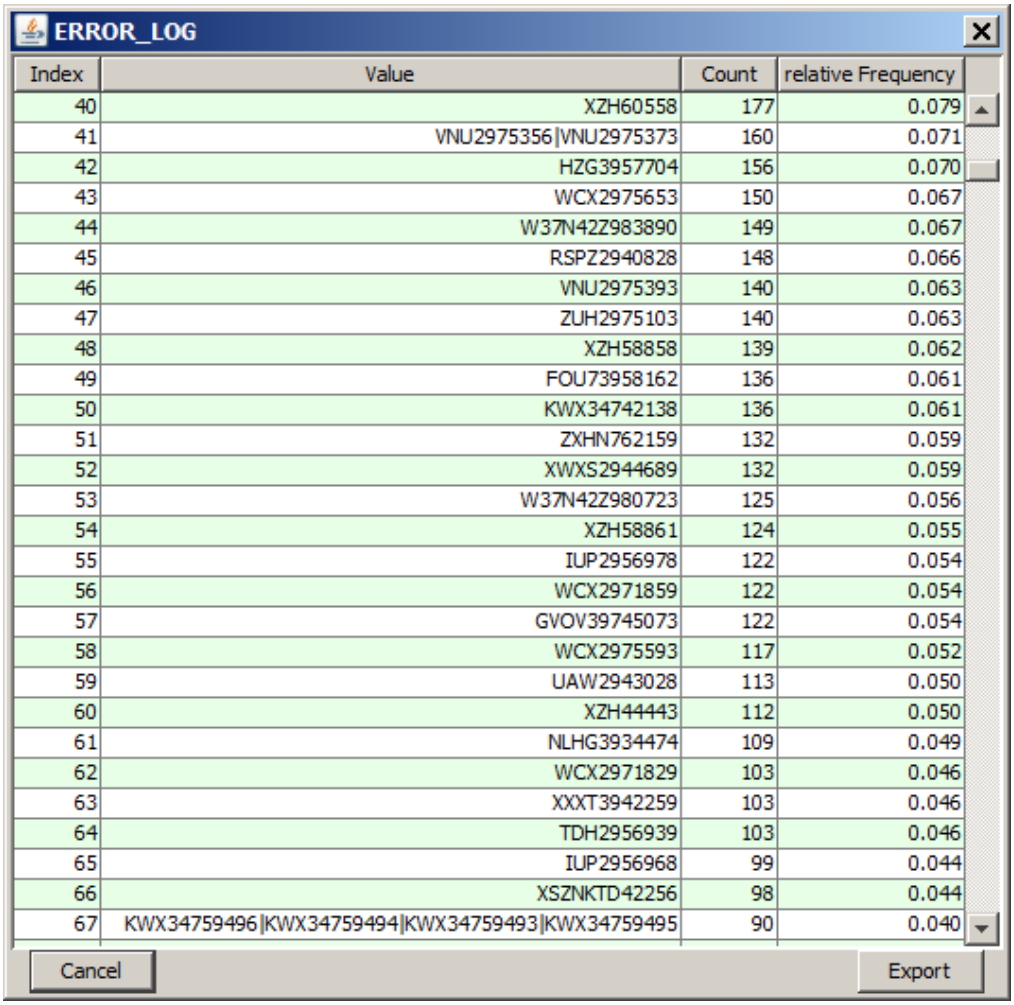


We choose the field `ERROR_LOG` and accept all other default settings in the window: search for value tupels whose single values appear in at least 10 data groups and for which at

least 95% of the data groups which contain one single value out of the tupel also contain the entire tupel. Then we press the **Start** button in order to start the tupel detection.

The screenshot printed above already shows the appearance of the window after the start command has been executed. 10 perfect tupels were found. The values forming these tupels were eliminated from the data, and whenever all values forming a tupel wer found in a data group, the tupel was inserted as a new single value into that data group. After replacing the single values by the tupels, there are 950 remaining different values in the field `ERROR_LOG`.

One can examine which values have been replaced by closing the 'perfect tupels' dialog and left-clicking on the histogram chart for the field `ERROR_LOG`: when scrolling through the value list we find new, longer error log codes which contain the concatenation character '|'. , for example the tupel of length 4 composed of the values `KWX34759496`, `KWX34759494`, `KWX34759493`, and `KWX34759495`. This tupel occurs in 90 repair cases.



3.2 The Module 'Correlations Analysis'

3.2.1 Purpose and short description

The data exploration module 'Correlations Analysis' serves to get an overview over the dependencies and correlations between the different data fields within a data source. This is done by creating a table of field-field contingency coefficients.

3.2.2 The tabular correlations view

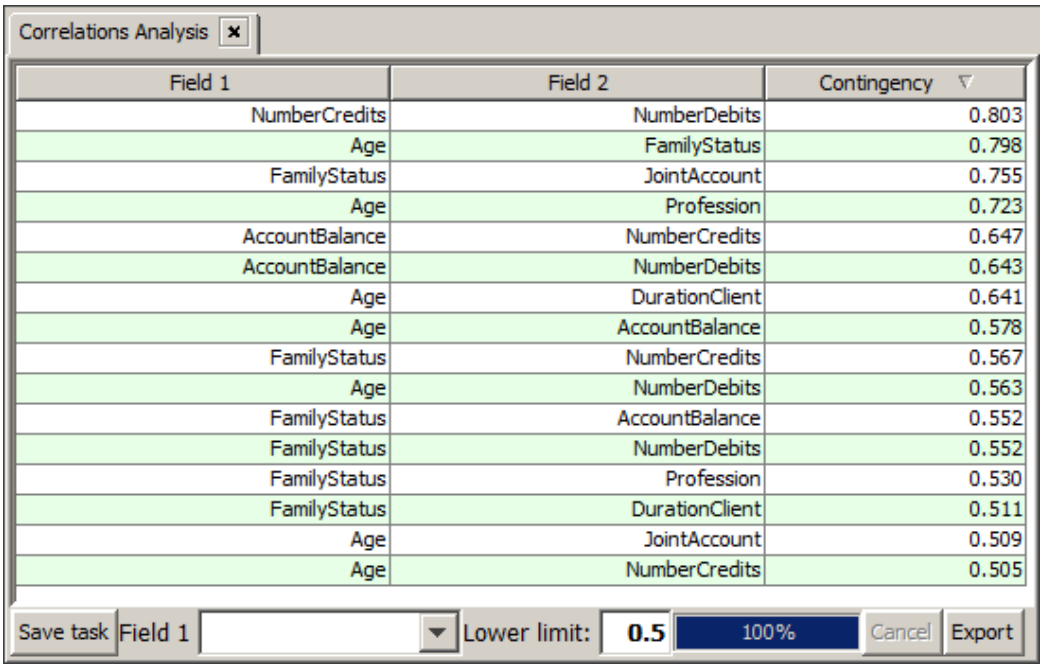
The main part of the correlations analysis panel displays a table of field-field correlations. In the literature, there are many different definition of measures for correlation, for example the linear correlations coefficient (or Pearson's correlations coefficient) between two nummeric data fields. Linear correlations coefficients have values between -1 (strong negative correlation) and +1 (strong positive correlation).

In Synop Analyzer, we use another measure for correlation which can also be calculated for pairs of textual data fields, or for a numeric and a textual field: the so-called adjusted contingeny coefficient C as it is defined in http://en.wikipedia.org/wiki/Contingency_table. This quantity assumes values between 0 (=no correlation) and 1 (=maximum correlation). The contingency coefficient is stronly related to the bivariate value-value matrix of the two involved fields as it is created in the Synop Analyzer module Bivariate Analysis: if one creates a bivariate value-value matrix for two data fields such that the field with the higher number of different values is traced on the y-axis, then one can derive from this matrix

- a contingency coefficient of 1 if and only of in each matrix row all cells except one are completely empty and all data records fall into one single populated cell. When displayed in the module Bivariate Analysis, the matrix would only show intensively red colored cells with `count=0` and one single intensively green colored cell per row.

- a contingency coefficient of 0 if and only if each matrix cell contains exactly the number of data records which one could have expected from calculating the product of the relative appearance frequencies of the column value and the row value. When displayed in the module Bivariate Analysis, such a matrix would have only white matrix cells.

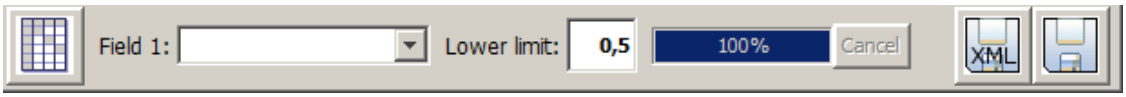
In the following we show an example for a continceny table. The example uses the sample data `doc/sample_data/customers.txt` and Synop Analyzer's default settings for importing the data.



By right-clicking with the mouse on one of the rows in the contingency table you open a new Bivariate Analysis panel in which the two data fields which appear in the selected table row have been chosen as the x-axis and the y-axis field.

3.2.3 The bottom tool bar

The tool bar at the lower border of the screen provides the following functions:



- :
  This button switches between the table view in the main panel and the alternative matrix view, which will be described in the next section of this chapter.

- **Field 1:**
  In this pop-up menu you can select the name of a data field from the data source, or you can select an empty string. If a field name has been selected, only contingency coefficients involving that field will be displayed. If no field has been selected, contingencies between all fields can be displayed.

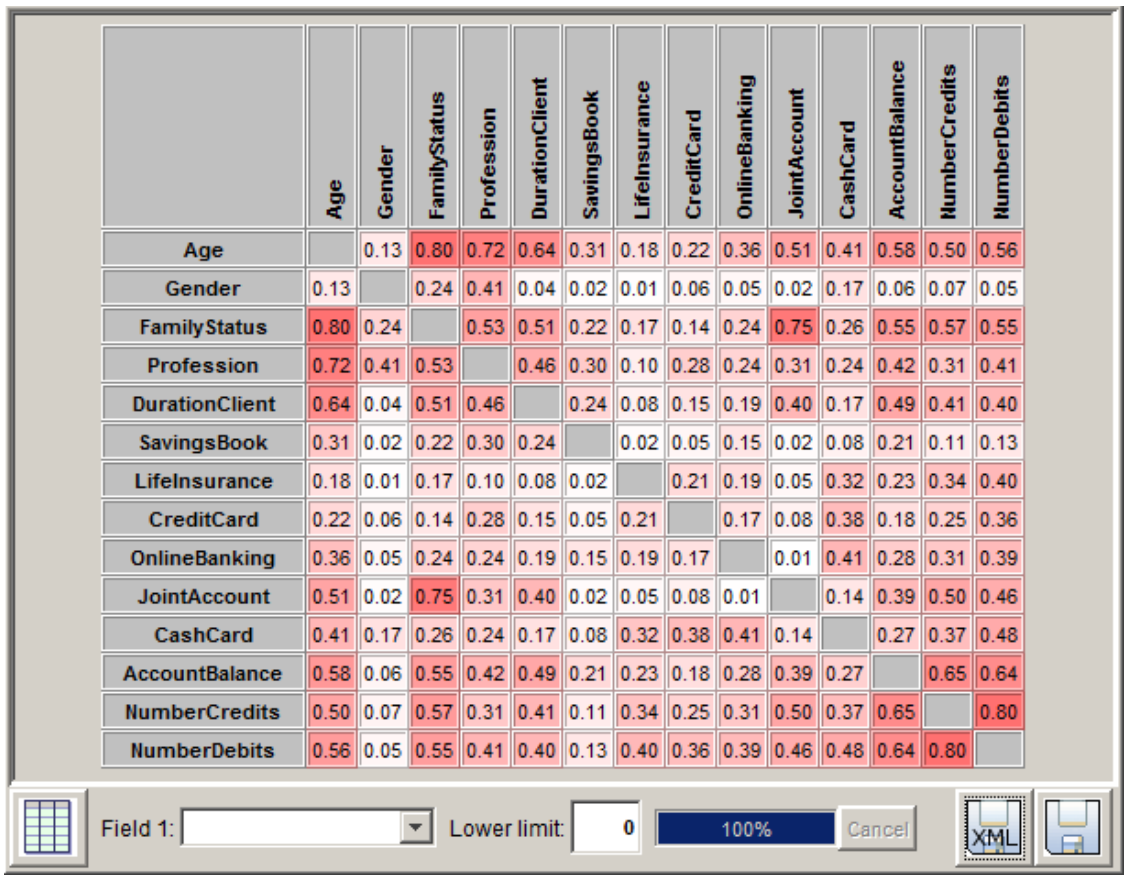
- **Lower limit:**
  Only contingency coefficients whose value is not below this threshold will be displayed.

- :
  By pressing this button, you can save the currently active data import settings and all settings performed in this module to a persistent XML parameter file. This file can later be opened via Synop Analyzer's main menu (**Analysis → Run Correlations Analysis**). In this way you can exactly reproduce the current data analysis screen without to be obliged to re-enter all settings and customizations.

- :
  Export the current data exploration results within this module into a spreadsheet in `.xlsx` format (MS-Excel© 2007+). The spreadsheet consists of one single worksheet which contains the tabular content of the main part of this module.

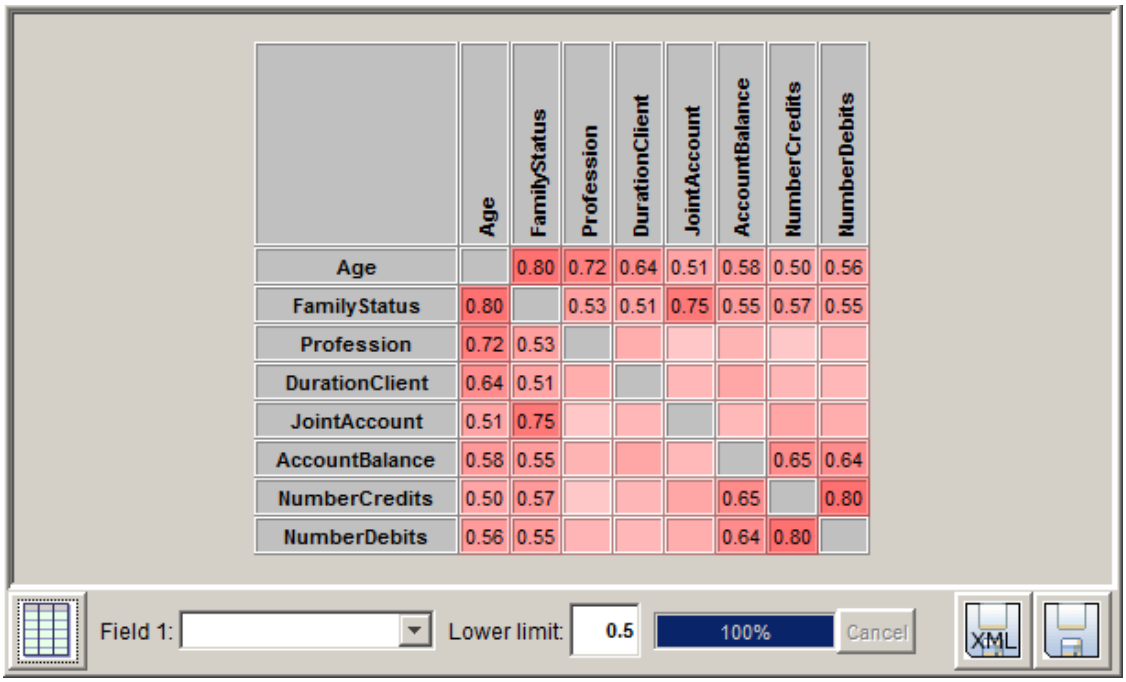
3.2.4 The correlations matrix view

In the matrix view, all field-field correlation numbers are shown in a compact matrix representation. The cells' background colors are the more intense the higher the correlation is.



If one chooses a minimum contingency threshold larger than zero in the toolbar, all correlation values smaller than this threshold are removed from the matrix. If a data

field has no correlation value above this threshold, the entire row and the entire column representing this field are removed from the matrix. This results in a more compact view which focusses on the highest correlations in the data.



3.3 The Module 'Bivariate Exploration'

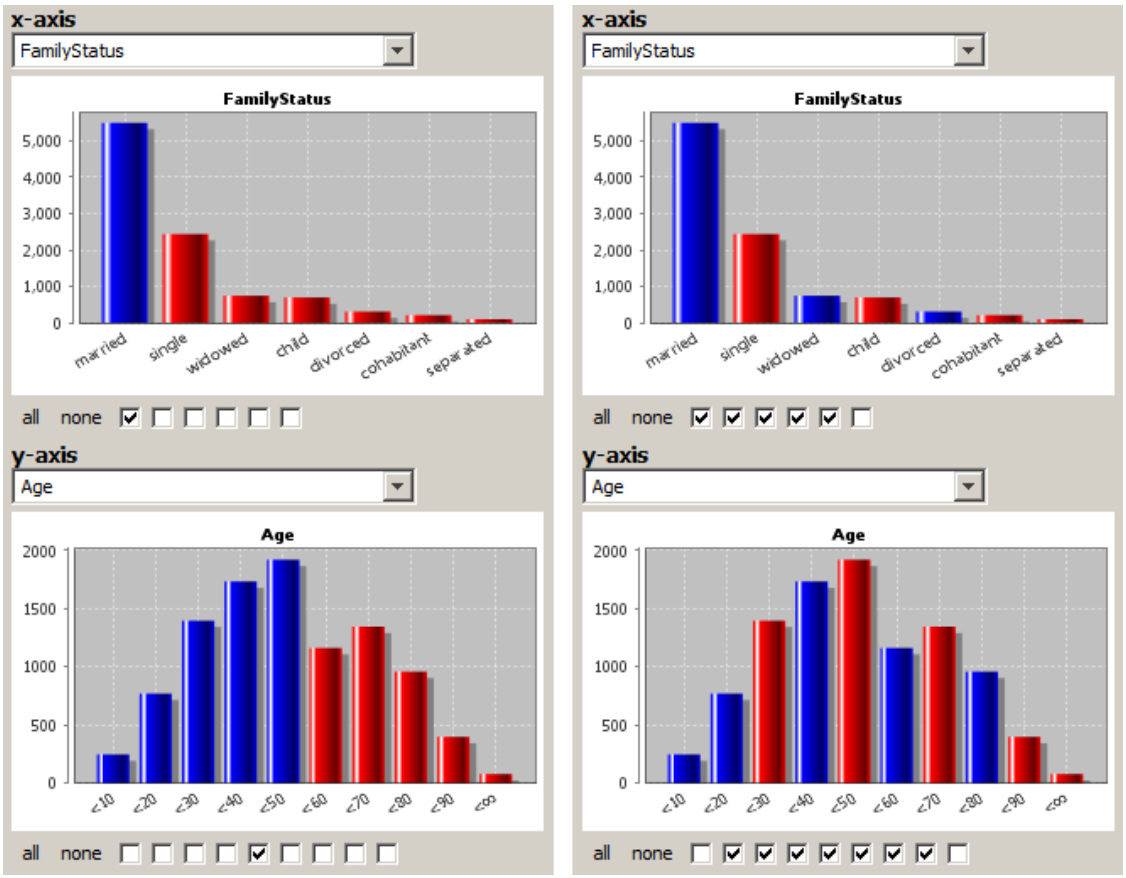
3.3.1 Purpose and short description

The data exploration module 'Bivariate Exploration' serves to study the dependencies and interrelations between the different values of two data fields in detail. This is done by creating a value combination matrix in which the values of the one field (the 'x-axis field') define the columns and the values of the other field (the 'y-axis field') define the matrix rows. A bivariate exploration can answer the following questions:

- Are there any correlations between the two fields, or are the values of the two fields statistically independent?
- If there are correlations, which values and value ranges are positively correlated and which 'repel' each other?
- Are there any combinations of values or value ranges which appear extremely less frequently than expected? This could be an indication for a data fault. Example: `FAMILY_STATUS=child` with `AGE>18`.
- How high is the absolute number of occurrences of certain combinations of values of the two fields?

3.3.2 The left hand panel: select fields and value ranges

In the left part of the module's screen window you can select the two data fields whose values are to be traced and whose interrelations are to be examined. This can be done by clicking on the 'arrow down' symbol at the right border of the white selection boxes below the head lines 'x-axis' and 'y-axis'. In the following screenshot, the sample data `doc/sample_data/customers.txt` has been imported into Synop Analyzer, and the two data fields `FamilyStatus` and `Age` have been selected as the two data fields on which a bivariate exploration is to be performed.



In the same screen part in which you select the data fields you also specify how fine-grained the values of the two data fields are to be treated in the bivariate analysis. This is done by selecting or deselecting some of the checkboxes below the histogram charts of the two data fields. Each checkbox stands for one possible value range split between two values or value ranges which are represented by one histogram bar in the chart above the checkbox. Therefore, the number of checkboxes is always the number of histogram bars minus one. Only if the check box is selected (marked), the corresponding range split is activated. Each color change between a red bar and a blue bar in the histogram above the check boxes represents one value range split. The neighbored values or value ranges whose histogram bars show the same color are considered one single value range within the bivariate analysis.

The left side of the figure above shows a rather 'coarse-grained' value range specification. On the x-axis, only the value `married`is separated from the other values, all remaining values are treated as one single value range. On the y-axis, we have set one single range split at the age of 50. That means, two value ranges will be created: `Age<50` and `Age`$\geq$`50`.

The right side of the figure above shows a more fine-grained value range specification. Almost all possible range splits have been set. Only some low-frequency values have been combined: on the x-axis the values `separated` and `cohabitant`, on the y-axis the value ranges `Age<10` and `Age=10..20` as well as `Age=80..90` and `Age`$\geq$`90`.

Accordingly, the biariate matrix resulting from the range specification on the left side is very small:

FamilyStatus Age	married	single, widowed,...	sum	χ² conf
<∞	2743 (+27%)	1195 (-33%)	3938	1.000
<50	2751 (-17%)	3311 (+21%)	6062	1.000
sum	5494	4506	10000	
χ² conf	1.000	1.000		1.000

whereas the bivariate matrix resulting from the range specification on the right side is much more detailed:

FamilyStatus Age	married	single	widowed	child	divorced	cohabitant, separated	sum	χ² conf
<∞	190 (-27%)	24 (-79%)	253 (+616%)	0 (-100%)	7 (-52%)	1 (-93%)	475	1.000
<80	608 (+16%)	72 (-69%)	256 (+259%)	0 (-100%)	15 (-49%)	7 (-77%)	958	1.000
<70	1024 (+39%)	118 (-64%)	159 (+59%)	0 (-100%)	33 (-21%)	10 (-76%)	1344	1.000
<60	921 (+44%)	113 (-60%)	34 (-61%)	0 (-100%)	69 (+92%)	24 (-34%)	1161	1.000
<50	1434 (+36%)	262 (-44%)	35 (-75%)	2 (-99%)	125 (+110%)	61 (+1%)	1919	1.000
<40	1041 (+9%)	511 (+21%)	4 (-97%)	2 (-98%)	57 (+6%)	117 (+115%)	1732	1.000
<30	276 (-64%)	1020 (+199%)	3 (-97%)	0 (-100%)	4 (-91%)	93 (+112%)	1396	1.000
<20	0 (-100%)	320 (+29%)	0 (-100%)	694 (+880%)	0 (-100%)	1 (-97%)	1015	1.000
sum	5494	2440	744	698	310	314	10000	
χ² conf	1.000	1.000	1.000	1.000	1.000	1.000		1.000

3.3.3 The bivariate matrix

The preceding section has described how a bivariate matrix such as the one in the figure above is generated. This sections will discuss which information can be derived from it.

- The 'Sum' column with gray background color indicates on how many data records (or data groups if a group field has been defined) the y-axis field has the value in the value range represented by the matrix row in which the 'sum' value is situated. For example, the number 475 in the first row of the 'Sum' column in the figure above indicates that in 475 data records the data field `Age` has a value ≥80. The 'Sum' row plays an analogous role for the x-axis field. The number 5494 in the leftmost field of the 'Sum' row in the figure above indicates that in 5494 data records the data field `FamilyStatus` has the value `married`.

- The number on the intersection of the sum row and the sum column - in the figure above the number '10000' - is the total number of data records (or data groups if a group field has been defined) on which the bivariate analysis was performed. If in the tool bar the checkbox **Ignore missing/invalid values** has not been checked, than this is the total number if data records or data groups in the input data. If the checkbox has been marked, it is the total number of data records or data groups on which both involved data fields have a valid value.

- The upper number in each pink or green colored matrix cell indicates the number of data records (or data groups if a group field has been defined) on which the corresponding combination of values of the two data fields occurs. For example, in the figure above, the number 190 in the pink matrix cell on the top left corner indicates that in 190 data records the field `Age` has a value in the range $\geq$80 and the field `FamilyStatus` has the value `married`.

- The second number in each matrix cell indicates how much the value of the upper number differs from its expected value. The expected value is the value which would arrise if the occurrence frequency of the combination of x-axis value and y-axis value occurred exactly as often as could be expected from the two values' occurrence probability. For example, the number '-27%' in the pink top-left matrix cell is the result of the following computation:

$$\texttt{N\_}_{expected} \texttt{ = 475/10000 * 5494/10000 * 10000 = 260.965}$$
$$\texttt{-27\% = (190 - 260.965) / 260.965.}$$

  The coloring of the cells' background is defined by the percentage number: the stronger below zero the more intensively red, the stronger above zero the more intensively green.
  In other words: pink and red cells represent combinations of values which occur unexpectedly rarely (negative correlation), green cells represent combinations of values which occur unexpectedly frequently (positive correlation).

- Each value in the 'χ^2 conf.' column with blue background color contains the statistical significance (confidence) of the differences between the expected and the actual occurrence frequencies in the matrix row in which the value is placed.
  In colloquial words: if the confidence value is larger than 0.95 (0.99, 1.000) then one can be 95% (99%, 100%) sure that the observed differences between actual and expected frequences are a statistically significant pattern and not random fluctuations.
  In mathematically precise words: the 'χ^2 conf.' value is the confidence level at which a χ^2 test with C-1 degrees of freedom and the following null hypothesis is rejected: "the actual occurrence frequencies in the matrix row have the same probability distribution as the expected occurrence frequencies." (Here, C is the number of matrix columns; in the figure above, C is 6.)
  An analogous definition holds for the 'χ^2 conf' values in the row with blue background color.

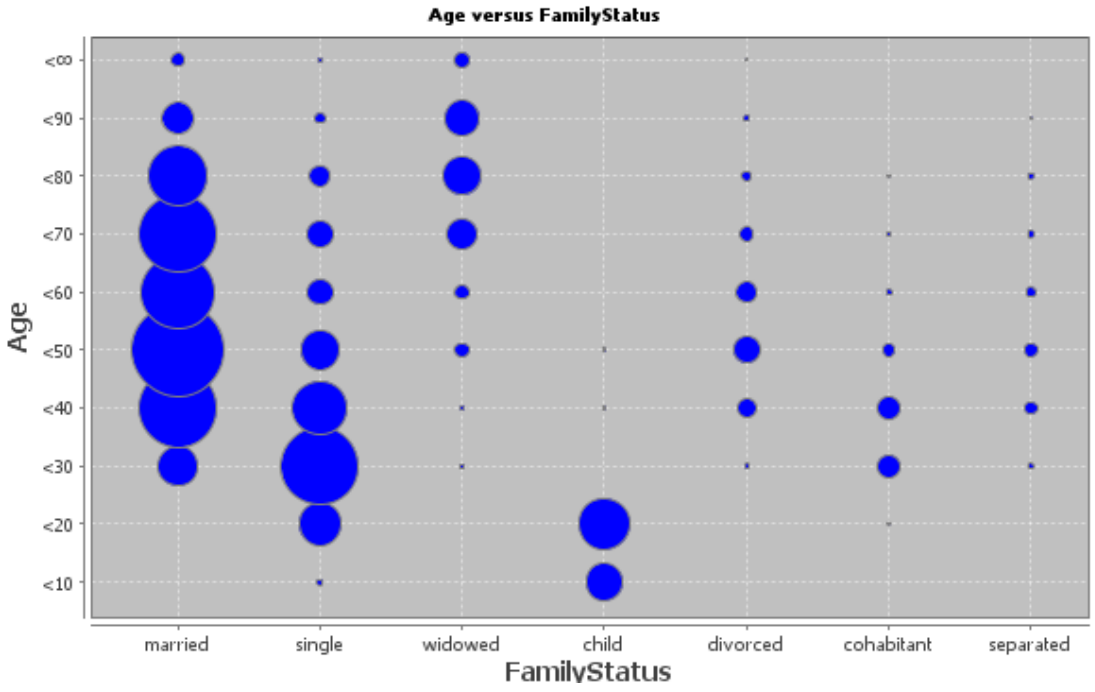
- The number at the intersection of the 'χ^2 conf' row and the '$\chi^2<$ conf' column contains the statistical significance level of the deviations between expected and actual

occurrence frequencies on the entire matrix.
In colloquial words: if the overall confidence value is larger than 0.95 (0.99, 1.000), one can be 95% (99%, 100%) sure that there is some statistically significant Korrelation between the two data fields, that means they are not statistically independent.
In mathematically precise words: the overall confidence number is the confidence level at which a χ^2 test with (R-1)*(C-1) degrees of freedom rejects the following null hypothesis: "The actual occurrence frequencies on the entire matrix have the same probability distribution as the theoretically expected occurrence frequencies." (Here, C is the number of matrix columns and R is the number of matrix rows).

3.3.4 The circle plot

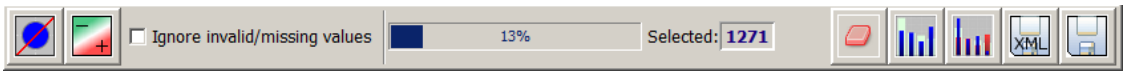
The bivariate matrix and the color scheme of its cells focus on visualizing relative differences between actual and expected frequencies of different combinations of values of the two involved fields. A second graphical visualizations of the interrelations between the two fields is given in the chart with the blue circles below the matrix. It displays the absolute size (measured in the number of data records respectively data groups) of the different possible combinations of field values. Each circle stands for one combination of field values, and the area of the circle is propoertional to the occurrence frequency.

From this plot one can understand very easily which combinations occur most frequently. On the other hand, also the most extremely untypical combinations can be detected quite easily in the form of little blue spot far away from any large circle in the same row or column of the plot. For example, the plot shown below contains two little blue dots in the column for the value `child` which are far above the typical age range of 0 to 20 years: these are children with ages between 30 and 50 years:



3.3.5 The bottom tool bar

The tool bar at the lower border of the screen provides the following functions:



- :
  By pressing this button, you can hide the circle plot whose blue circles (disks) show the absolute population sizes of all possible value combinations of the two selected data fields.
- :
  By pressing this button, you can invert the red-green color scheme in the bivariate matrix. Per default, value combinations (matrix cells) which appear more frequently than expected are colored green, combinations which appear less frequently than expected are colored red. If the quantity counted within the cells represents something negative, e.g. cost or error cases, it is often more intuitive that larger counts are colored red (problem hot spots) and smaller counts are colored green (less error-prone cases).
- **Ignore missing/invalid values:**
  If this checkbox is not marked, all data records will be used and counted when creating the bivariate matrix. If the checkbox is not marked, only the data records which have valid values in both involved fields are being counted.
- **Selected:**
  The absolute and relative number of currently selected data records (or data groups if a group field has been selected).
- :
  Deletes all selections of matrix cells (which are signaled by blue frames).
- :
  Starts a multivariate exploration of the data records in the currently selected cells of the bivariate matrix. See section Multivariate exploration of selected matrix cells.
- :
  Starts a split analysis. The data records in the currently selected cells of the bivariate matrix are the test group, all other data records form the control group.
- :
  By pressing this button, you can save the currently active data import settings and all settings performed in this module to a persistent XML parameter file. This file can later be opened via Synop Analyzer's main menu (**Analysis** → **Run Bivariate**

Exploration). In this way you can exactly reproduce the current data analysis screen without to be obliged to re-enter all settings and customizations.

- :
  Eport the current data exploration results within this module into a spreadsheet in `.xlsx` format (MS-Excel© 2007+). The spreadsheet contains several worksheets: one with png graphics of the two charts on the right side of the bivariate exploration panel, one with the bivariate matrix in the form of an editable, sortable worksheet. And if some bivariate matrix cells have been selected, there are two more sheets containing the selected data records in tabular form as well as a multivariate exploration of these records compared to the entire data.

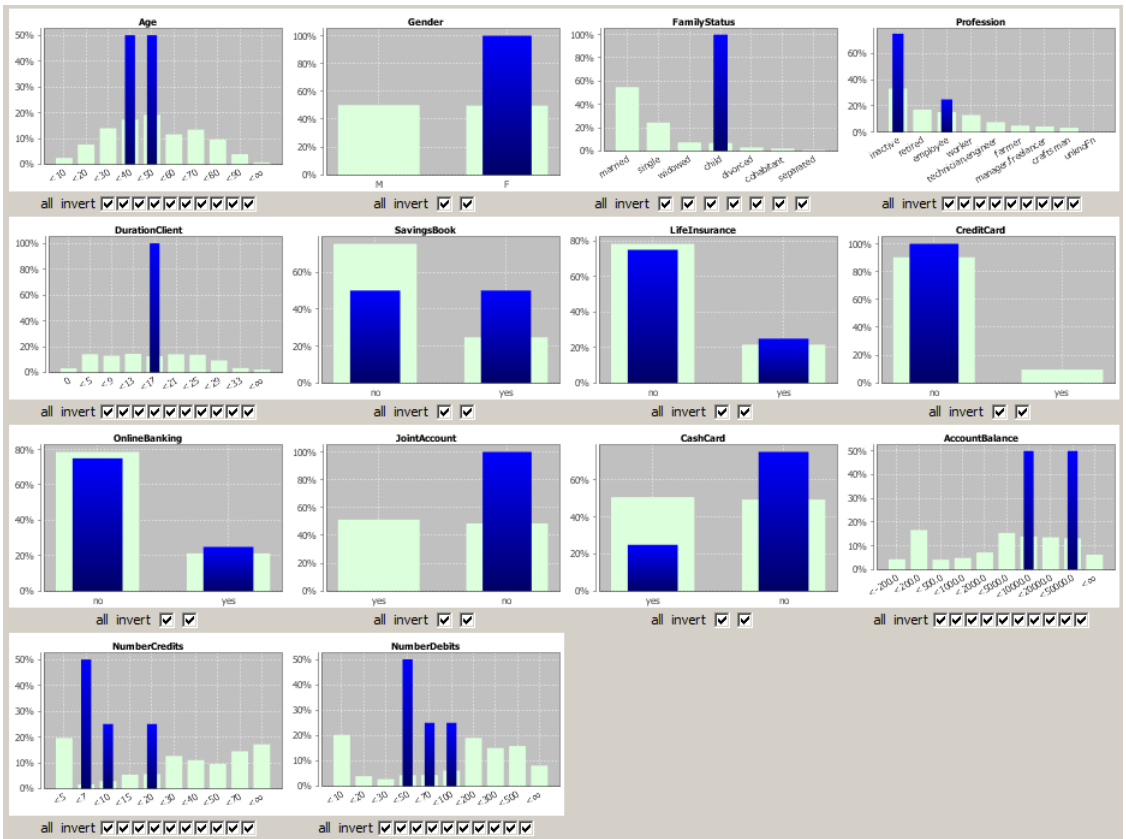
3.3.6 Selecting and exploring matrix cells

By clicking with the left mouse button one can select a cell of the bivariate matrix. If you keep the <CTRL> key pressed during mouse-clicking, you can select several matrix cells. Once one or more cells have been selected, the bottom tool bar of the bivariate analysis panel shows the total number of data records (or data groups if a group field has been specified) in the selected cells. By clicking the button you can open a new multivariate exploration panel inwhich the value distributions of the selected data records (or data groups) are compared to the value distributions on the entire data.

We want to demonstrate this with the help of the example which has been shown above: the bivariate matrix showing the interrelations between the data fields `Age` and `Family Status` from the sample data `doc/sample_data/customers.txt`. In this matrix we have selected two noticeable cells, presumable data errors: children at ages between 30 and 50 years.

FamilyStatus Age	married	single	widowed	child	divorced	cohabitant, separated	sum	χ^2 conf
<∞	190 (-27%)	24 (-79%)	253 (+616%)	0 (-100%)	7 (-52%)	1 (-93%)	475	1.000
<80	608 (+16%)	72 (-69%)	256 (+259%)	0 (-100%)	15 (-49%)	7 (-77%)	958	1.000
<70	1024 (+39%)	118 (-64%)	159 (+59%)	0 (-100%)	33 (-21%)	10 (-76%)	1344	1.000
<60	921 (+44%)	113 (-60%)	34 (-61%)	0 (-100%)	69 (+92%)	24 (-34%)	1161	1.000
<50	1434 (+36%)	262 (-44%)	35 (-75%)	2 (-99%)	125 (+110%)	61 (+1%)	1919	1.000
<40	1041 (+9%)	511 (+21%)	4 (-97%)	2 (-98%)	57 (+6%)	117 (+115%)	1732	1.000
<30	276 (-64%)	1020 (+199%)	3 (-97%)	0 (-100%)	4 (-91%)	93 (+112%)	1396	1.000
<20	0 (-100%)	320 (+29%)	0 (-100%)	694 (+880%)	0 (-100%)	1 (-97%)	1015	1.000
sum	5494	2440	744	698	310	314	10000	
χ^2 conf	1.000	1.000	1.000	1.000	1.000	1.000		1.000

The multivariate epxploration of the four data sets from these two cells shows that most probably the age is correct by the family status is outdated, since the other properties of these data records, for example the account balance or the elevated accounting activity are more typical for adults than for children.



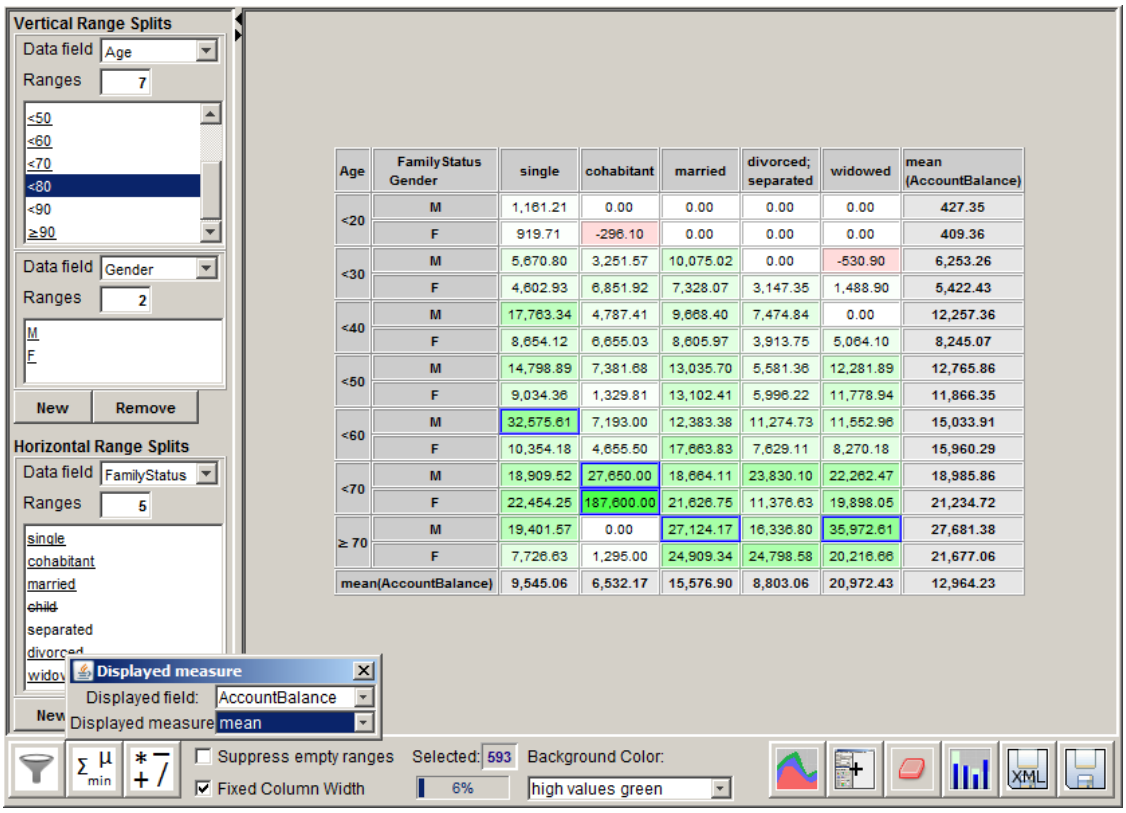
3.4 The Module 'Pivot Tables'

3.4.1 Purpose and short description

The data exploration module 'Pivot Tables' serves to study the dependencies and interrelations between the different values of several data fields in a tabular view. The module can be considered a functionally enlarged variant of the Bivariate Exploration module, offering the following additional features:

- The value ranges which define the horizontal and the vertical dimension of the displayed table can come from more than one data field, and there are more degrees of freedom as to joining, rearranging and dropping certain data field ranges.
- There are more choices as to which quantity is displayed in the main cells of the table. Bivariate Exploration always displays the number of data records or data groups. In Pivot Tables, we can also display certain statistical measures of further data field, such as mean, minimum, maximum or the field's value sum.
- There are several different coloring schemes which define the color-coding of the background of the pivot table cells.
- You can pre-filter the data which enter into the analysis.
- You can connect two pivot tables by computation formulas.
- You can transform the table view into a chart view.

The screenshot below shows a sample application: the table displays the mean account balances of bank customers traced by age, gender and family status. The ranges which have been marked with a blue line are the ranges in which the average account balance is at least twice the mean account balance on all customers.



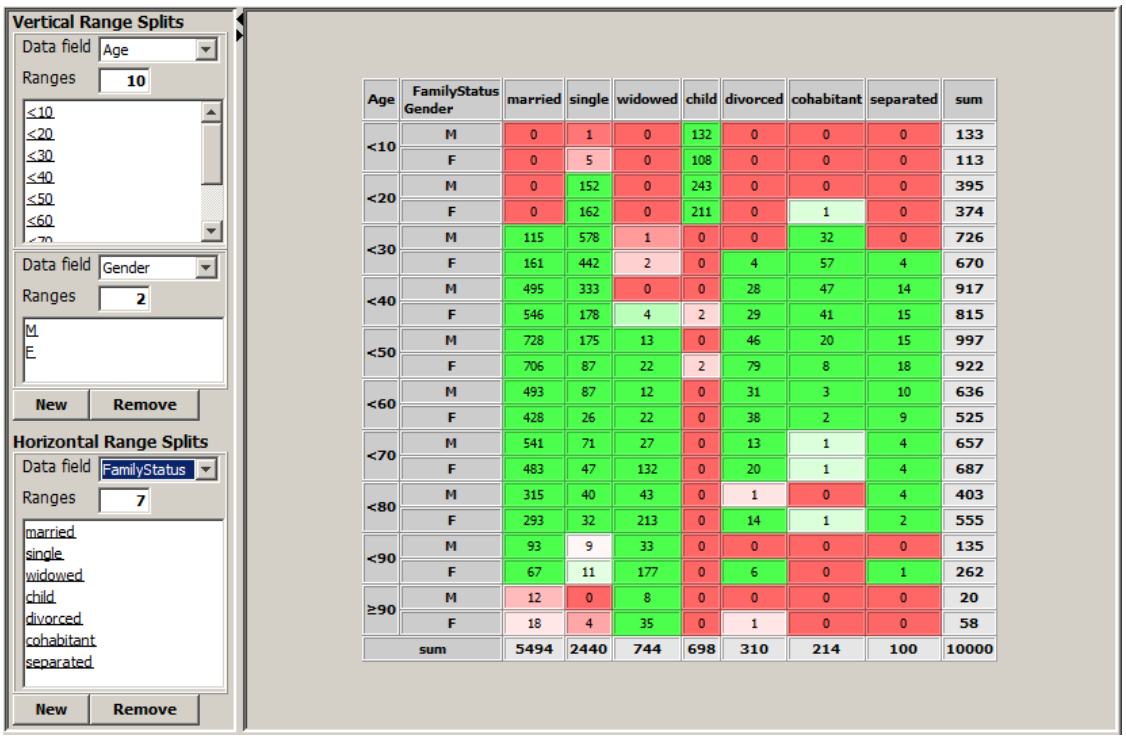
Age	Family Status / Gender	single	cohabitant	married	divorced; separated	widowed	mean (AccountBalance)
<20	M	1,161.21	0.00	0.00	0.00	0.00	427.35
	F	919.71	-296.10	0.00	0.00	0.00	409.36
<30	M	5,670.80	3,251.57	10,075.02	0.00	-530.90	6,253.26
	F	4,602.93	6,851.92	7,328.07	3,147.35	1,488.90	5,422.43
<40	M	17,763.34	4,787.41	9,668.40	7,474.84	0.00	12,257.36
	F	8,654.12	6,655.03	8,605.97	3,913.75	5,064.10	8,245.07
<50	M	14,798.89	7,381.68	13,035.70	5,581.36	12,281.89	12,765.86
	F	9,034.36	1,329.81	13,102.41	5,996.22	11,778.94	11,866.35
<60	M	32,575.61	7,193.00	12,383.38	11,274.73	11,552.96	15,033.91
	F	10,354.18	4,655.50	17,663.83	7,629.11	8,270.18	15,960.29
<70	M	18,909.52	27,650.00	18,664.11	23,830.10	22,262.47	18,985.86
	F	22,454.25	187,600.00	21,626.75	11,376.63	19,898.05	21,234.72
≥ 70	M	19,401.57	0.00	27,124.17	16,336.80	35,972.61	27,681.38
	F	7,726.63	1,295.00	24,909.34	24,798.58	20,216.66	21,677.06
mean(AccountBalance)		9,545.06	6,532.17	15,576.90	8,803.06	20,972.43	12,964.23

3.4.2 The left hand panel: select fields and value ranges

In the left part of the module's screen window you can select the data fields and the field value ranges which will define the rows and the columns of the pivot table.

The buttons **New** below the headlines **Vertical Ranges** and **Horizontal Ranges** create a new range split for either the x- or the y-axis of the resulting table, each range split being based on one single data field. After pressing the button, a new range specification window appears in the left screen column. In the selector box **Data Field** you select the data field on which the range split will be based.

The screenshot printed below shows the pivot table which results from choosing the vertical range split fields `Age` and `Gender` and the horizontal range split field `FamilyStatus` on the data `doc/sample_data/customers.txt`. This pivot table is very similar to the bivariate matrix created by the module 'Bivariate Exploration' for the two data fields `Age` and `FamilyStatus`.



Age	FamilyStatus / Gender	married	single	widowed	child	divorced	cohabitant	separated	sum
<10	M	0	1	0	132	0	0	0	133
	F	0	5	0	108	0	0	0	113
<20	M	0	152	0	243	0	0	0	395
	F	0	162	0	211	0	1	0	374
<30	M	115	578	1	0	0	32	0	726
	F	161	442	2	0	4	57	4	670
<40	M	495	333	0	0	28	47	14	917
	F	546	178	4	2	29	41	15	815
<50	M	728	175	13	0	46	20	15	997
	F	706	87	22	2	79	8	18	922
<60	M	493	87	12	0	31	3	10	636
	F	428	26	22	0	38	2	9	525
<70	M	541	71	27	0	13	1	4	657
	F	483	47	132	0	20	1	4	687
<80	M	315	40	43	0	1	0	4	403
	F	293	32	213	0	14	1	2	555
<90	M	93	9	33	0	0	0	0	135
	F	67	11	177	0	6	0	1	262
≥90	M	12	0	8	0	0	0	0	20
	F	18	4	35	0	1	0	0	58
sum		5494	2440	744	698	310	214	100	10000

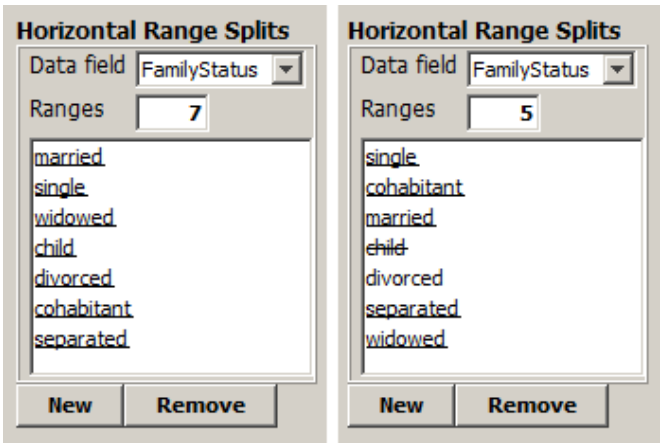
Each range split can be modified by mouse actions on the list area showing the single field values. There are three different display modes for each list entry, each display mode representing a certain usage mode of the corresponding field value.

- If the field value is underlined, this means that after the field value a new value range, that means a new table row or column, begins.
- If the field value is striked through, this means that the field value and all data records containing this value are suppressed in the pivot table and do not contribute to the displayed aggregation value shown in the table cells.
- If the field value is neither underlined nor striked through, this means that the data records containing the value contribute to the table, but they form a single table row or column together with the folling value in the list.

The following mouse actions are supported on the list area:

- A left mouse click on one of the list entries removes or adds a range split, that means it toggles between underlined and normal display mode.
- A mouse click with the middle or right mouse button on one of the list entries activates or deactivates the corresponding field value, that means it toggles between striked throgh and normal display mode.
- By drawing with the mouse (that means by keeping the left mouse button pressed while moving a list entry to a new position within the list) you can rearrange the field value order of textual data fields. Numeric fields, on the contrary, have an inherent natural ordering ('smaller than'), therefore a reordering makes no sense here.

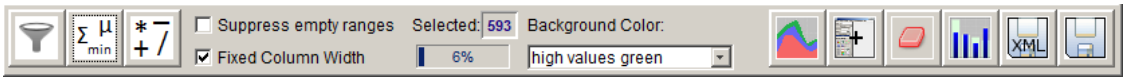
The picture displayed below shows on its left side the default state of the range split definition window for the data field `FamilyStatus` of the data file `doc/sample_data/-customers.txt`. On the right side, the picture shows a user-modified state.



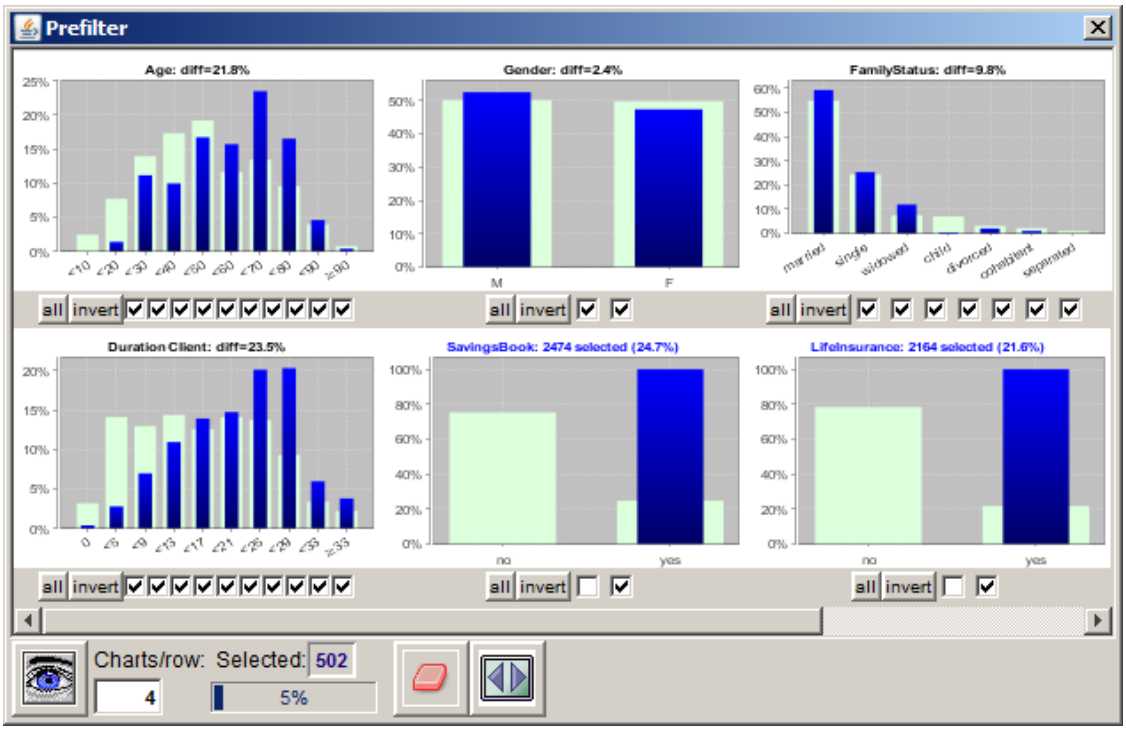
The default state creates the pivot table with 7 horizontal ranges shown at the beginning of this section. The nodified state creates a pivot table such as the one shown in the introductory section of this chapter. That table has only 5 horizontal ranges. The value `child` has been suppressed, the values `divorced` and `separated` have been combined into one single range, and the ranges have been reordered into the 'logical' order: `single` before `cohabitant` before `married` before `separated; divorced` before `widowed`.

3.4.3 The bottom tool bar

The tool bar at the lower border of the screen provides the following functions:

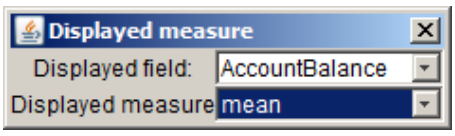


- :
  This button opens a pop-up window in which a data prefiltering can be defined. The prefiltering restricts the set of data records which will enter into the pivot tables. It is performed on a multivariate exploration panel view in which the data field values to be filtered out can be deselected.



The screenshot shown above displays an example for a data prefiltering: it filters out all data records representing customers who do not have a savings book or who di not have a life insurance.

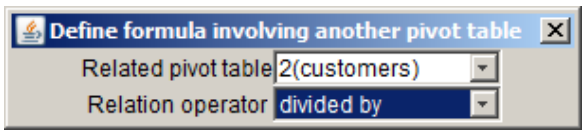
- $\Sigma^{\mu}_{\min}$:
Via this button you can specify the measure to be displayed in the pivot table. Per default, the number of data records or data groups is displayed in the pivot table. Using the two selection boxes **Displayed field** and **Displayed measure** in the pop-up dialog, you can tell the pivot table to display a statistical measure of a selected numeric data field instead, for example one of the quantities mean, sum, minimum or maximum.



In the screenshot shown above, the mean account balance has been chosen as the measure to be displayed.

- $*\,\overline{+}\,/$:
Here you can specify a second pivot table. The current pivot table's values will then be submitted to a mathematical operation (addition, subtraction, multiplication or division) with the corresponding table cell values of the second pivot table. Eligible for being chosen as second table are all currently opened pivot tables whose number of rows respectively columns is either 1 or equal to the number of rows respectively

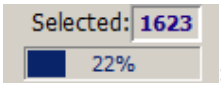
columns of the current pivot table.



In the screenshot displayed above, the pivot table in the second currently opened pivot table panel for the data source `customers` has been selected as the related table. The specified computation operation for the relation is **divided by**. That means, in the current pivot table, the value of each numeric table cell will be divided by the value of the corresponding cell in the pivot table **2(customers)** before it is displayed on screen.
A sample application scenario of that feature is calculating failure rates (isochronous lines) in technical quality monitoring. Assume that we have created a pivot table which traces failure counts as a function of production period (table rows) and usage time (table columns). If we then create a second pivot table which traces production numbers as a function of production period (table rows), we can relate our first pivot table to the second one using the computation operator **divided by**. The resulting pivot table (or its resulting chart view) then shows isochronous failure rate lines.

- **Suppress empty ranges:**
  If this checkbox is marked, all columns and rows of the pivot table will be removed which only contain the value 0.

- **Fixed Column Width:**
  If this checkbox is marked, columns of the pivot table will have the same fixed width. If the checkbox is not marked, each column only has the minimum required width for displaying all its content.

- 

:
  The progress bar and the text field **Selected** show the size of the currently selected subset of the data: the number in the progress bar is the percentage of the entire data; the number to the right of the **Selected** label is the absolute number of data records (or data groups if a group field has been specified) in all currently selected cells of the pivot table.
  Left clicking with the mouse on the progress bar or the output field showing the number of selected data groups opens a pop-up window which shows the currently applied selection criteria in the form of a `SQL SELECT` statement. By pressing a button in the pop-up window, you can copy this statement into the system clipboard and insert it from there into a SQL script which you can then deploy on your database management system.

- **Background Color:**
  By means of this selection box you can specify the coloring scheme for the background of the pivot table cells. Besides the neutral white background mode there are two modes which color-code the absolute size of the number in the table cell

('high values green' and 'high values red'), and two modes which are similar to the color-coding of the Bivariate Exploration module and which measure the difference between the actual and the expected value of the table cell.

- :
  Opens a pop-up window in which a chart representation of the current pivot table can be created. The chart representation will be explained in detail in the last section of this chapter.

- :
  Creates a new in-memory data source which represents the current content of the pivot table. That means, the data fields of the new data source are the column header names of the pivot table, and the data records are the numeric-valued rows of the pivot table (except the final summary row, which is ignored). The new data source will be displayed in a newly created tab in the left column of the Synop Analyzer workbench; it can be used for arbitrary new analysis steps.

- :
  Deletes all selections of table cells (which are signaled by blue frames).

- :
  Starts a multivariate exploration of the data records in the currently selected cells of the pivot table.

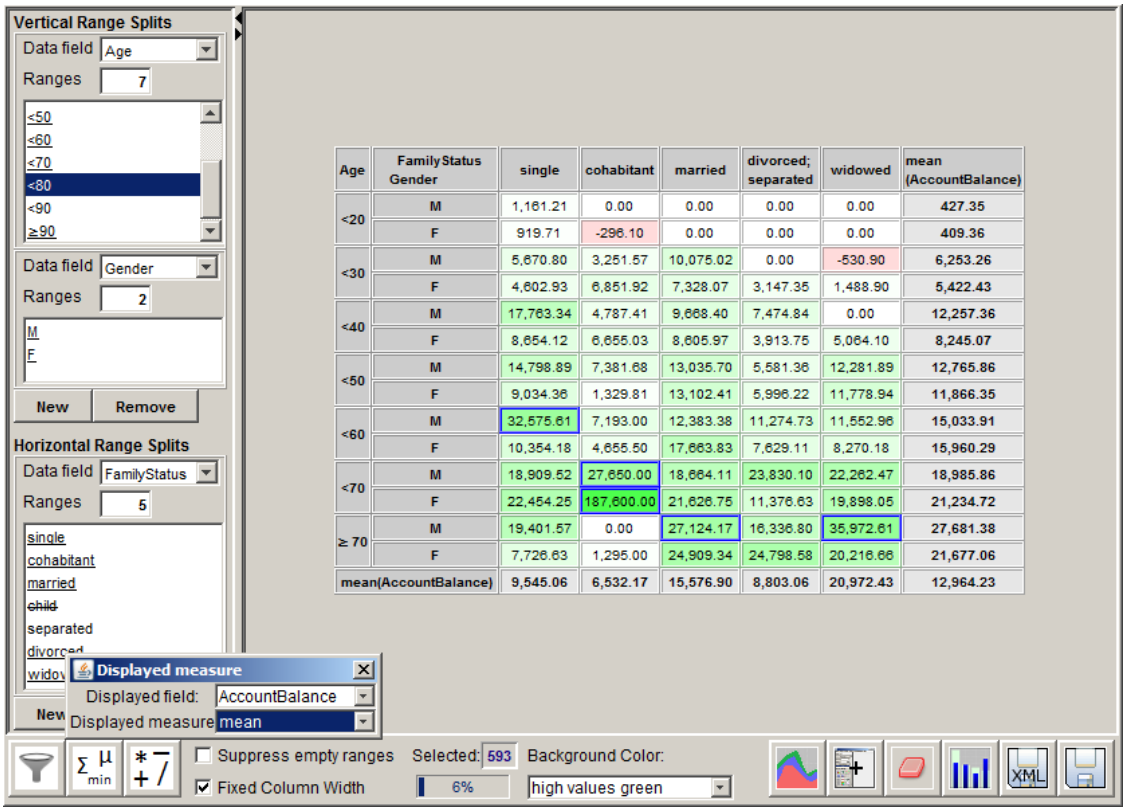
-
  :
  By pressing this button, you can save the currently active data import settings and all settings performed in this module to a persistent XML parameter file. This file can later be opened via Synop Analyzer's main menu (**Analysis → Run Pivot Table**). In this way you can exactly reproduce the current data analysis screen without to be obliged to re-enter all settings and customizations.

- :
  Exports the current data exploration results within this module into a spreadsheet in `.xlsx` format (MS-Excel© 2007+). The spreadsheet contains several worksheets: one with png graphics of the two charts on the right side of the bivariate exploration panel, one with the bivariate matrix in the form of an editable, sortable worksheet. And if some bivariate matrix cells have been selected, there are two more sheets containing the selected data records in tabular form as well as a multivariate exploration of these records compared to the entire data.

3.4.4 The pivot table panel

The preceding sections have described how a pivot table view can be generated and modified using the left-hand panel and the bottom toolbar. The resulting table is displayed in the main part of the panel.

The screenshot below shows a sample application: the table displays the mean account balances of bank customers from file `doc/sample_data/customers.txt`traced by age, gender and family status. The ranges which have been marked with a blue line are the ranges in which the average account balance is at least twice the mean account balance on all customers.



Age	Family Status / Gender	single	cohabitant	married	divorced; separated	widowed	mean (AccountBalance)
<20	M	1,161.21	0.00	0.00	0.00	0.00	427.35
	F	919.71	-296.10	0.00	0.00	0.00	409.36
<30	M	5,670.80	3,251.57	10,075.02	0.00	-530.90	6,253.26
	F	4,602.93	6,851.92	7,328.07	3,147.35	1,488.90	5,422.43
<40	M	17,763.34	4,787.41	9,668.40	7,474.84	0.00	12,257.36
	F	8,654.12	6,655.03	8,605.97	3,913.75	5,064.10	8,245.07
<50	M	14,798.89	7,381.68	13,035.70	5,581.36	12,281.89	12,765.86
	F	9,034.36	1,329.81	13,102.41	5,996.22	11,778.94	11,866.35
<60	M	32,575.61	7,193.00	12,383.38	11,274.73	11,552.96	15,033.91
	F	10,354.18	4,655.50	17,663.83	7,629.11	8,270.18	15,960.29
<70	M	18,909.52	27,650.00	18,664.11	23,830.10	22,262.47	18,985.86
	F	22,454.25	187,600.00	21,626.75	11,376.63	19,898.05	21,234.72
≥ 70	M	19,401.57	0.00	27,124.17	16,336.80	35,972.61	27,681.38
	F	7,726.63	1,295.00	24,909.34	24,798.58	20,216.66	21,677.06
mean(AccountBalance)		9,545.06	6,532.17	15,576.90	8,803.06	20,972.43	12,964.23

The table contains the following rows and columns:

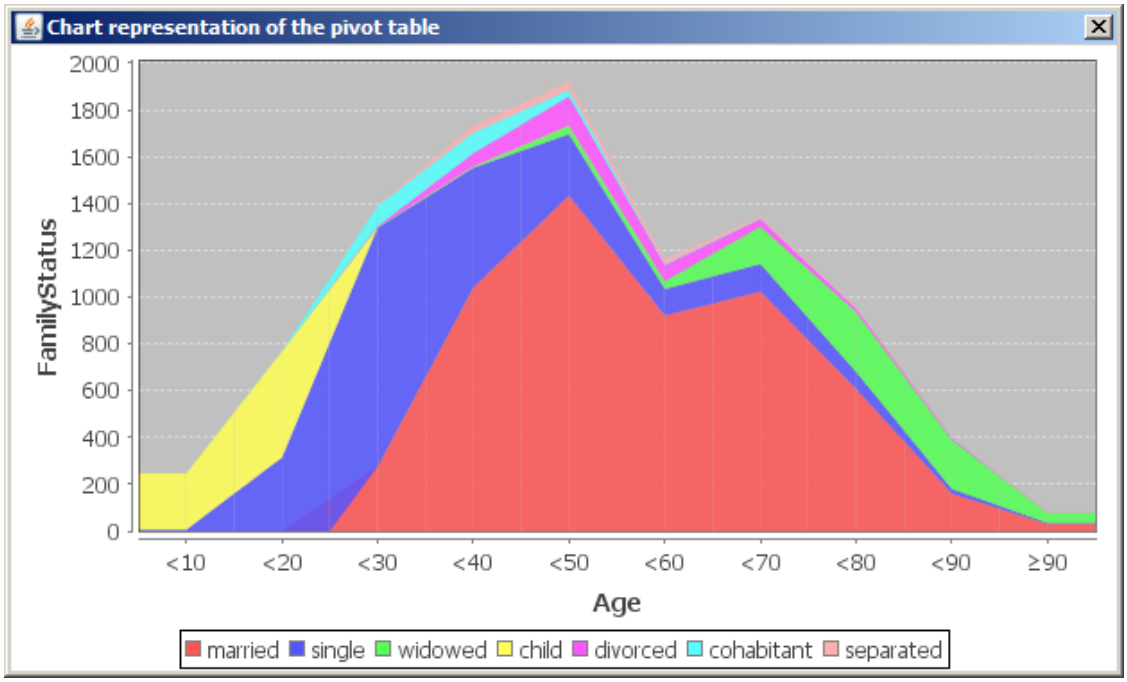
- One or more header rows and columns with gray background contain the range selection criteria, that means the data fields and their value ranges which define the specific table row or columns.
- The table cell in the lower right corner contains the value of the measure to be displayed in the table on the entire data. In the screenshot shown above, it is the number 12964.32, the average account balance of all 10000 customers.
- The last column and the last row of the table display the mean value of the measure displayed in the table on the data subset specified by the corresponding header column or row. For example, the number 9545.06 in the second cell of the last

row indicates that the mean account balance of the singles among the customers is 9545.06.

- The remaining table cells (with white, pink or green background) show the value of the measure to be displayed in the table on the data records which form the intersection of the data subsets representing the table row and the table column.

3.4.5 The chart panel

Using the toolbar button you can create a chart representation of the pivot table's content. The row names of the pivot table will become the labels on the chart's x-axis, the chart's y-axis will show the values displayed in the numeric pivot table cells. Each column name of the pivot table defines a uniquely colored area in the chart. The areas are stacked above each other so that the upper end of the chart represents the number printed in the rightmost table cell of the pivot table.



The chart shown above displays the chart representation of the pivot table printed in section 'The left column' which has customers' `Age` ranges as rows and customers' `FamilyStatus` as columns.

3.5 The Module 'Multivariate Exploration'

3.5.1 Purpose and short description

The data exploration module 'Multivariate Exploration' serves to study the dependencies and interrelations between the different values of several data fields in detail. To this purpose, the module displays histogram charts of all (or a user-defined subset of all) data fields on a single screen panel. By mouse-clicking the user can interactively select an deselect values and value ranges in an arbitrary combination of the histograms, thereby defining a multivariate data selection. The modifications in the field value distributions of all data fields which result from the selection are displayed in real time on screen even on very large data.

3.5.2 Understanding the main panel

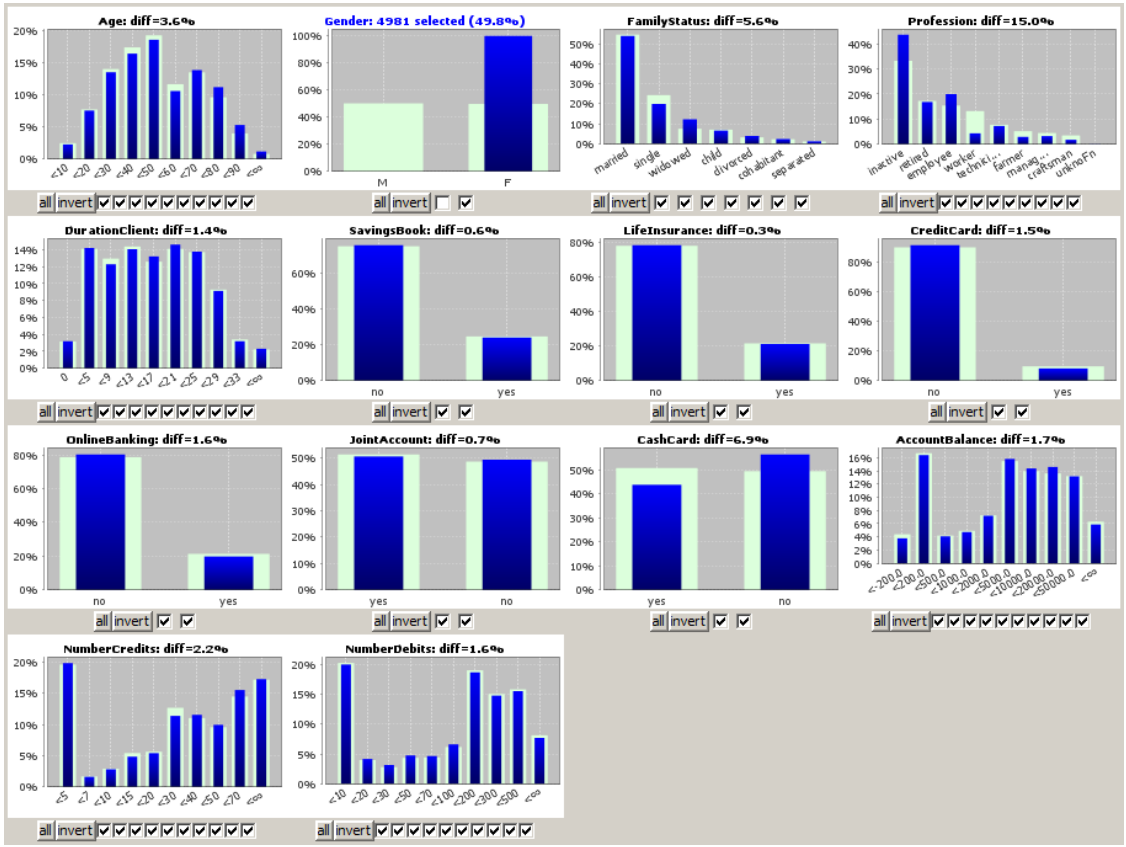
The main part of the Multivariate Exploration panel consists of one histogram chart per active data field. Each histogram chart compares a field's value distribution on the currently selected subset (blue bars) to the field's value distribution on the entire data (light green bars).

Histograms with more than 36 bars cover the entire screen width, histograms with not more than 18 bars are grouped into tupels of N charts per screen row, where N is the number entered into the tool bar input field named **Charts/row**. If this input field contains the value 0, the software decides autonomously how many charts to put into one screen row. Charts with 19 to 36 bars occupy twice as much horizontal space as the charts with not more than 20 bars. In order to avoid ugly gaps in the arrangements of the charts on screen, the 'large' charts (those with more than 18 bars) are placed before the 'small' charts, that means those with less than 19 bars.

In the histogram charts for non-numeric data fields, the values are arranged by descending occurrence frequency from left to right.If a data field has more then N different values, where N is the number in the input field **#values (text fields)** in the Input Data panel, then only the N most frequent values have been separately recorded when the data were imported. All other values have been summarized into the 'rest' value 'others'. This rest value will be represented in the chart by one single bar with label 'others'. If there is no such 'rest' value in the data, it can still be the case that there are so many different values that it is impossible to draw a histogram bar for each of them. In this case, the histogram chart will be truncated after 80 bars (you can change that value of 80 in the pop-up dialog **Preferences → Multivariate Preferences**). The fact that some bars could not be displayed is indicated by an additional label saying "... ?? others", where ?? is the number of suppressed bars.

Numeric data fields - such as the field `Age` in the picture below - often have so many different values that a binning into a small number of value ranges or intervals is reasonable. The number of bins and the bin boundaries have been defined and can be modified in the Input Data Panel.

By clicking on one of the checkboxes which are situated below each chart, a value selection (restriction) can be defined for the corresponding data field. In the following screenshot, the sample data `doc/sample_data/customers.txt` have been imported into Synop Analyzer. Then, the Multivariate Exploration module has been started and the left checkbox below the chart for the field `Gender` has been deselected. That means, we have removed the male customers from the blue selected data. Hence, the latter represent the female customers, and the differences between the light green and the blue bars represent the differences between the female customers and all customers.

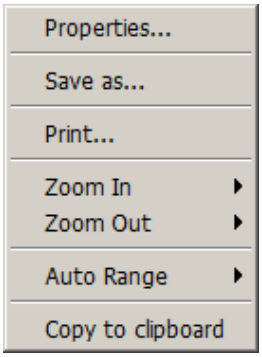


We derive from the picture that the professions of the female customers strongly differ from those of the male customers - more women are employees or inactive whereas much more men are workers - while there is almost no difference between both groups as to the possession rate of savings books, life insurances or credit cards.

The user can now interactively select an deselect values and value ranges in one or more arbitrary other data fields, thereby defining a multivariate data selection. The calculation of the overall selection is performed on an in-memory representation of the data which is optimized for those multivariate 'slicing' operations over several fields. Therefore, the results can be calculated and displayed within fractions of a second even on multi-gigabyte data.

By drawing with the mouse (keep the left mouse button pressed while moving) on a histogram chart you mark a rectangular region in which you want to zoom in.

By right-clicking on a histogram chart you open the pop-up dialog shown below. In this dialog, you can modify the appearance of the histogram chart (text fonts and sizes, axis styles, labels, etc.) via the menu item **Properties**. You can also save the chart as PNG graphics, print it or copy it as png graphics object to the system clipboard.

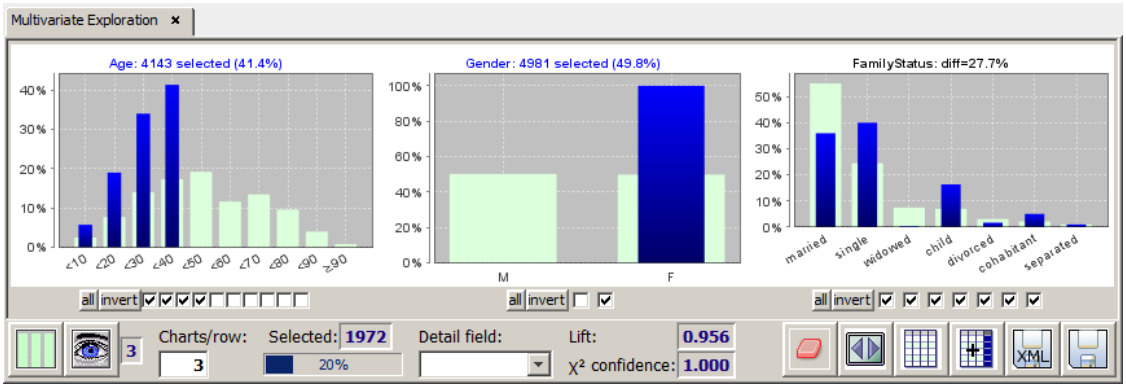


Using the button **Visible fields** in the bottom toolbar, you can hide and remove certain fields from the charts panel in order to get a clearly arranged picture on data with many data fields.

3.5.3 Working with the range selector buttons

Now we want to study the possibilites of selecting and deselecting value ranges by means of the button bars below the histogram charts in more detail. To that purpose we focus on a part of the screenshot shown above, namely the histograms and button bars for the three data fields `Age`, `Gender` and `FamilyStatus`.

In addition to the existing range limitation on the field `Gender` we want to restrict the values of the field `Age`, namely we want to focus on the customers below 40 years. To that purpose we could deselect the six rightmost checkboxes under the histogram for field `Age`. A bit faster is the alternative approach of deselecting the four leftmost checkboxes and then clicking on the **invert** button. The **invert** button inverts the existing range selection on a data field. The button **all**removes all ranges restrictions from the field.



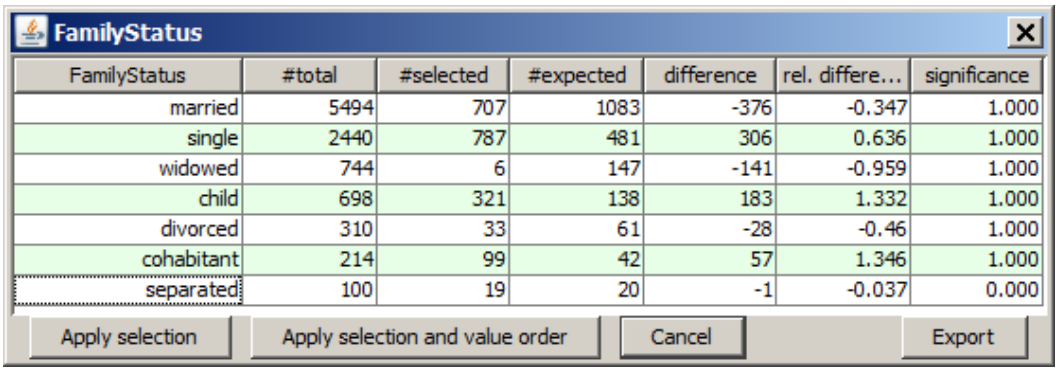
The new selection defines 4143 customers in the selected `Age` region. As the intersection with the existing preselection of 4981 female customers we get 1972 or about 20% young female customers (these numbers are displayed in and next to the progress bar in the bottom tool bar).

The range restriction in the field `Age` instantaneously changes the heights of the blue bars in all other data fields. As expected, the percentage of children and singles in the field `FamilyStatus` have grown significantly. The difference between the the selected subset and the light green background distribution on the entire data has grown strongly on most data fields. The displayed '**diff** value is calculated as the total length of all parts of the blue bares which exceed the light green bars divided by the total length of all blue bars (the latter is always 100% if the respective field is not set-valued).

The chart titles of the fields in which we have specified a range restriction (selection) are displayed in blue; the titles of the 'response' fields in which the observed differences between blue and light green bars are a reaction of range selections in other fields are displayed in black.

3.5.4 Working with detail pop-up dialogs für single fields

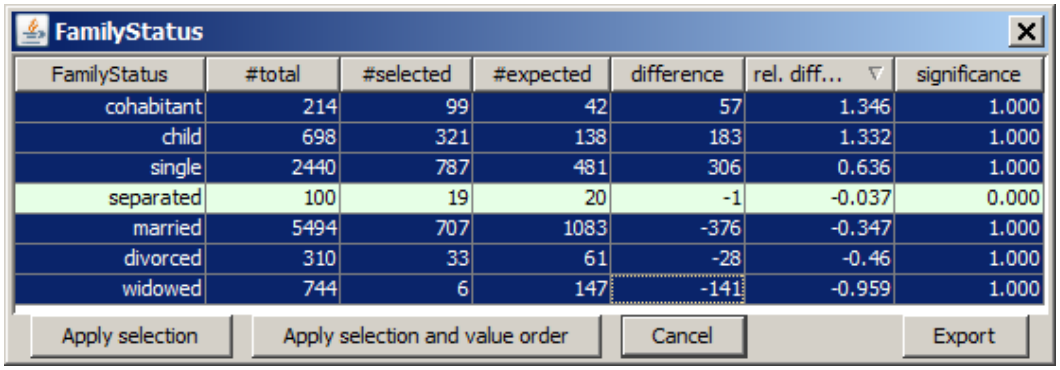
A left mouse-click on one of the histogram charts opens a tabular detail statistics which shows the field's values or value ranges and their actual and expected occurrence frequencies on the selected data. **#expected** is the expected number of selected data records under the assumption that the value's relative frequency on the selected data is identical to the value's relative frequency on the entire data. The columns **difference** and **rel. difference** contain the absolute and relative difference between the actual and the exected occurrence frequency. Finally, the column **significance** displays the result of a χ^2 significance test which indicates whether the observed difference between actual and expected occurrence frequencies on the selected data are statistically significant (significance values close to 1) or not (significance values below 0.95...0.9).



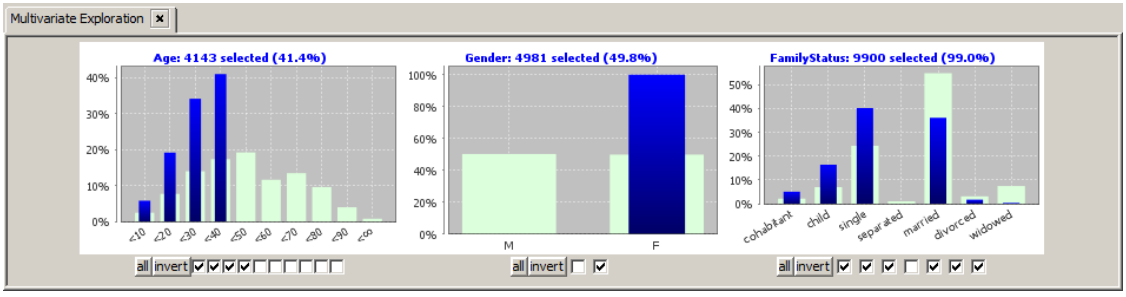
If a non-numeric data field has many different values, for example far more than 100, then the available space in the histogram is not sufficient for displaying a separate bar and checkbox for each of them. In this case, the pop-up detail view is the only possibility for seeing all different values and for selecting or deselecting single values which do not

figure among the 80 most frequent values. This selection or deselection can be performed by mouse-clicks on certain table rows in the detail view. If you keep the <CTRL> key pressed while clicking, you can select more than one row; by keeping the <SHIFT> key pressed you can select an entire value range. After selecting the desired table rows you activate your selection and close the pop-up view by pressing the button **Apply selection**.

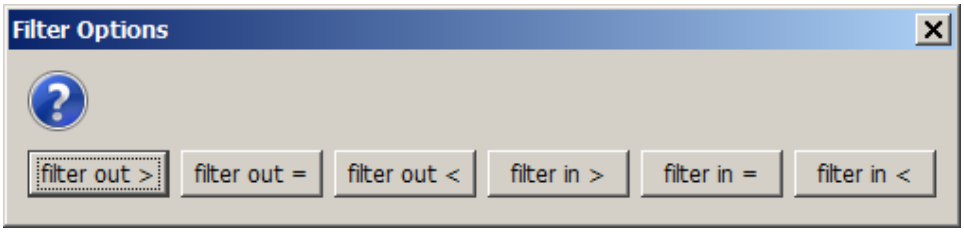
In the details pop-up view you can also reorder the values by pressing on one of the column heads. This sorts the values ascendingly or descendingly by the values of the clicked column. Repeated clicks invert the sorting order. In the screenshot shown below, we have sorted by descending relative difference. This brings the value `cohabitant` to the top position. Then we have deselected the value on which the actual frequency does not significantly differ from the expected frequence, namely the value `separated`.


FamilyStatus
FamilyStatus | #total | #selected | #expected | difference | rel. diff... | significance
cohabitant | 214 | 99 | 42 | 57 | 1.346 | 1.000
child | 698 | 321 | 138 | 183 | 1.332 | 1.000
single | 2440 | 787 | 481 | 306 | 0.636 | 1.000
separated | 100 | 19 | 20 | -1 | -0.037 | 0.000
married | 5494 | 707 | 1083 | -376 | -0.347 | 1.000
divorced | 310 | 33 | 61 | -28 | -0.46 | 1.000
widowed | 744 | 6 | 147 | -141 | -0.959 | 1.000
Apply selection
Apply selection and value order
Cancel
Export

If we now leave the pop-up window by pressing the button **Apply selection and value order**, both the new value ordering and the value selection is applied to the histogram chart:

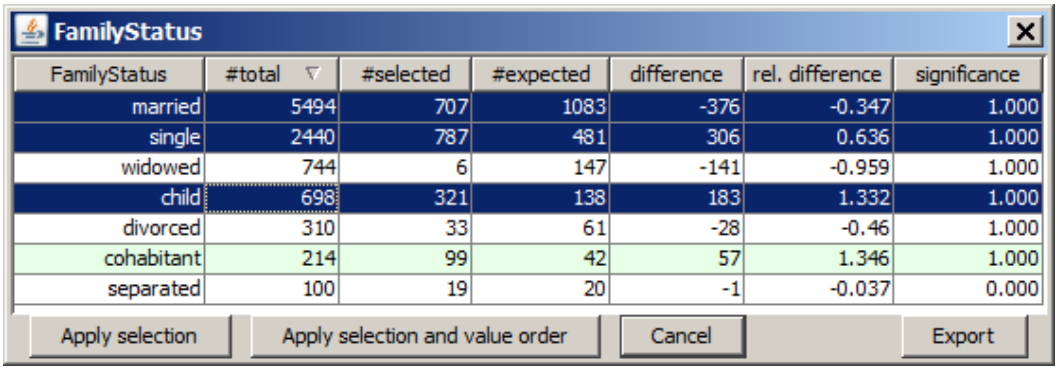


The details pop-up view offers yet another feature: if you right-click on one of the table cells, the following options dialog pops up:



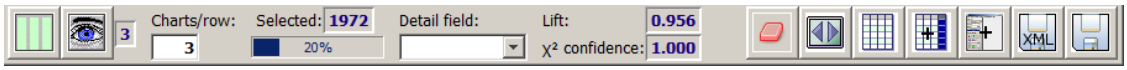
This dialog permits selecting or deselecting all table rows whose values in the column in which the click was performed are in a certain value range, and this selection can be performed by one single click. This is an enormous reduction of effort especially if the field contains hundreds or thousands of different values.

The following picture results from right-clicking on the value `321` in the column `#selected` and by choosing the option **deactivate <** in the options dialog. This choice deselects all table rows which have a value of less than `321` in the column `#selected`.


FamilyStatus
FamilyStatus | #total | #selected | #expected | difference | rel. difference | significance
married | 5494 | 707 | 1083 | -376 | -0.347 | 1.000
single | 2440 | 787 | 481 | 306 | 0.636 | 1.000
widowed | 744 | 6 | 147 | -141 | -0.959 | 1.000
child | 698 | 321 | 138 | 183 | 1.332 | 1.000
divorced | 310 | 33 | 61 | -28 | -0.46 | 1.000
cohabitant | 214 | 99 | 42 | 57 | 1.346 | 1.000
separated | 100 | 19 | 20 | -1 | -0.037 | 0.000
Apply selection
Apply selection and value order
Cancel
Export

3.5.5 The bottom toolbar

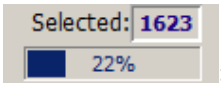
The tool bar at the lower screen border provides the following buttons and functions:



- :
  Toggle the histogram display mode. This button opens a pop-up panel in which the chart type (histogram bar chart, line chart or area chart) and the display mode can be selected. In the default display mode, the sum of all light-green background bar heights is 100% ('sum mode').In the option mode, the so-called 'single mode', each single light-green background bar is rescaled to 100%. This second mode is particularly useful for studying the relative frequency differences between the selected data and the overall data on the various values or value ranges of a data field.

- :
  Via this button you open a pop-up dialog which permits to hide certain data fields from the histogram chart panel. This feature is described in more detail in section Rearranging and suppressing fields. The blue number to the right of the **Visible fields** button shows the total number of remaining visible fields.

- **Charts/row**
  In this input field you can specify how many of the 'normal' histogram charts with not more than 18 bars should be put into one single screen row. The smaller the number, the larger will be each single histogram chart.

- 

  :
  The progress bar and the text field **Selected** show the size of the currently selected subset of the data: the number in the progress bar is the percentage of the entire data; the number to the right of the **Selected** label is the absolute number of selected data records (or data groups if a group field has been specified).
  Left clicking with the mouse on the progress bar or the output field showing the number of selected data groups opens a pop-up window which shows the currently applied selection criteria in the form of a `SQL SELECT` statement. By pressing a button in the pop-up window, you can copy this statement into the system clipboard and insert it from there into a SQL script which you can then deploy on your database management system.
  Right clicking with the mouse on the progress bar or the output field showing the number of selected data groups opens a pop-up window which serves to deactivate all value ranges in all visible data fields in which the currently selected data subset is empty or significantly under-represented. The number entered into the pop-up window's input field defines the minimum degree of under-representation required for deselecting a value range. The predefined default value is 0.33. That means, all histogram bars in which the blue bar's height is less than one third of the green bar's height will be deactivated.

- **Detail field:**
  By means of the selection box named 'Detail field' you can specify one data field whose value distribution within each single histogram bar representing the selected data will be graphically displayed using different colors instead of the uniform blue bar color. More information on this feature can be found in section Working with detail structure fields.

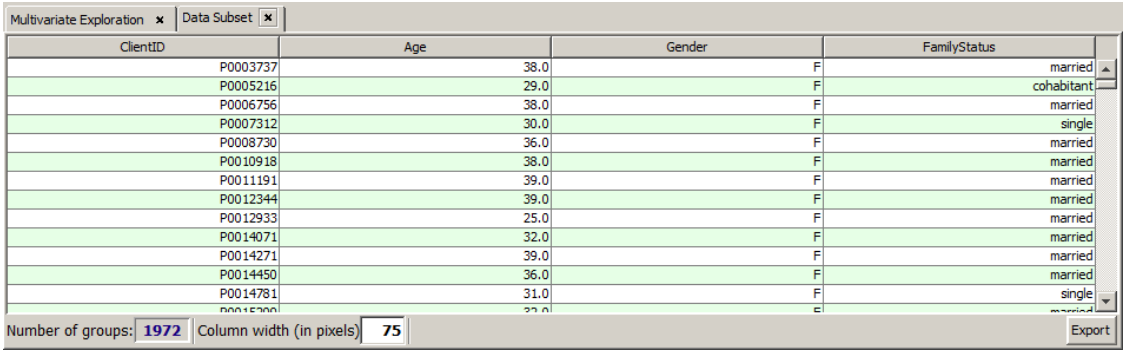
- **Lift:**
  The text field **Lift** indicates whether the combination of field value ranges defining the current selection 'attracts' or 'repulses' each other. A lift value of 1.0 indicates that the different selected field value ranges are statistically independent: lift values larger than 1.0 (less than 1.0) indicate that the different selected value ranges occur more (less) frequently together than expected in the case of statistical independence.

- χ^2 **Confidence:**
  The text field χ^2 **conf.** contains the statistical confidence that the selected subset differs significantly from the entire data in at least one data field's value distribution. More formally spoken, the value is the confidence level with which the hypothesis «The currently selected subset has the same value distribution in all data fields as the entire data»is rejected by a χ^2 significance test.

- :
Undo all range restrictions; select all data records.

- :
By clicking on this button you re-draw all histogram charts, thereby adapting their size to the current screen width.

- :
This button opens a new panel which contains the currently selected data records in tabular form. In the panel, you can sort the selected data by any data field and export the extire selection or a subset into a flat file or spreadsheet.



- :
This button appends a new two-valued (Boolean) data field to the data. The new field represents the current selection: it contains '1' for all data records or data groups which are contained in the current selection, and '0' for those ones not contained in the current selection. You can specify the name if the new field in a pop-up dialog which opens up after pressing this button.

- :
This button transforms the current selection of data records or data groups and the currently visible data fields into a new data source within Synop Analyzer. The new data source is automatically opened as a separate tab in the left column of Synop Analyzer workbench. You can then apply all Synop Analyzer analysis modules to this new data.

- :
By pressing this button, you can save the currently active data import settings and all settings performed in this module to a persistent XML parameter file. This file can later be opened via Synop Analyzer's main menu (**Analysis → Run Multivariate Exploration**). In this way you can exactly reproduce the current data analysis screen without to be obliged to re-enter all settings and customizations.

- :
  Export the current data exploration results within this module into a spreadsheet in `.xlsx` format (MS-Excel© 2007+). The spreadsheet contains several worksheets: one with a single PNG graphics for each histogram chart, one with a single PNG graphics for all charts, a data sheet which contains the selected data records, and one more worksheet for each detail pop-up window which ever has been opened by mouse-clicking on one of the histogram charts.

3.5.6 Rearranging and suppressing fields

Clicking on the button **Visible fields** opens a pop-up dialog in which the following actions can be performed:

- Hide certain data fields so that no histogram is displayed for them. You can hide a field by left-clicking the field name while keeping the <CTRL> key pressed.
- Rearrange the histograms on screen: if you draw a field name with the mouse to another vertical position and release the left mouse key there, the field name is moved to the new position. Note: moving a field name is only possible within its 'group'. The data fields with many different values and large histogram charts form the first group, the fields with normally sized charts form the second group.
- Sort the data fields with respect to a user-selected filter criterion. The pull-down menu named **Sort by** at the lower border of the **Visible fields** pop-up dialog makes it possible to sort and reorder the displayed histograms with respect to a couple of sorting criteria. The meaning of the criteria **lexical field order**, **field order in the data** and **correlation with detail field** should be evident. The criterion **rel. difference** sorts the fields on which a manual range restriction has been defined at first place and then the other fields sorted by descending `diff` value.The criterion χ^2 **conf** also places the fields with manual range restrictions in front, followed by the other fields sorted by decreasing χ^2 confidence value. The χ^2 confidence value indicates the level of confidence of the assertion that the value distribution of the blue selected data significantly differs from the value distribution of the light green overall data. In general, this criterion has some similarity and correlation with the criterion **rel. difference**. However, a small relative difference of, say, 1% can be very significant on a field with many data records only very few different field values, whereas a relative difference of 10% can be non-significant on a field with many different values and few data records.
- Exchange the quantitative difference measure shown in the charts' titles. After selecting the option **Sort by** → χ^2 conf</but>, in the **Visible fields** dialog, the chart titles display the difference measure χ^2 `conf`. Sorting by **rel. differeence** switches back to displaying the relative difference (`diff`).

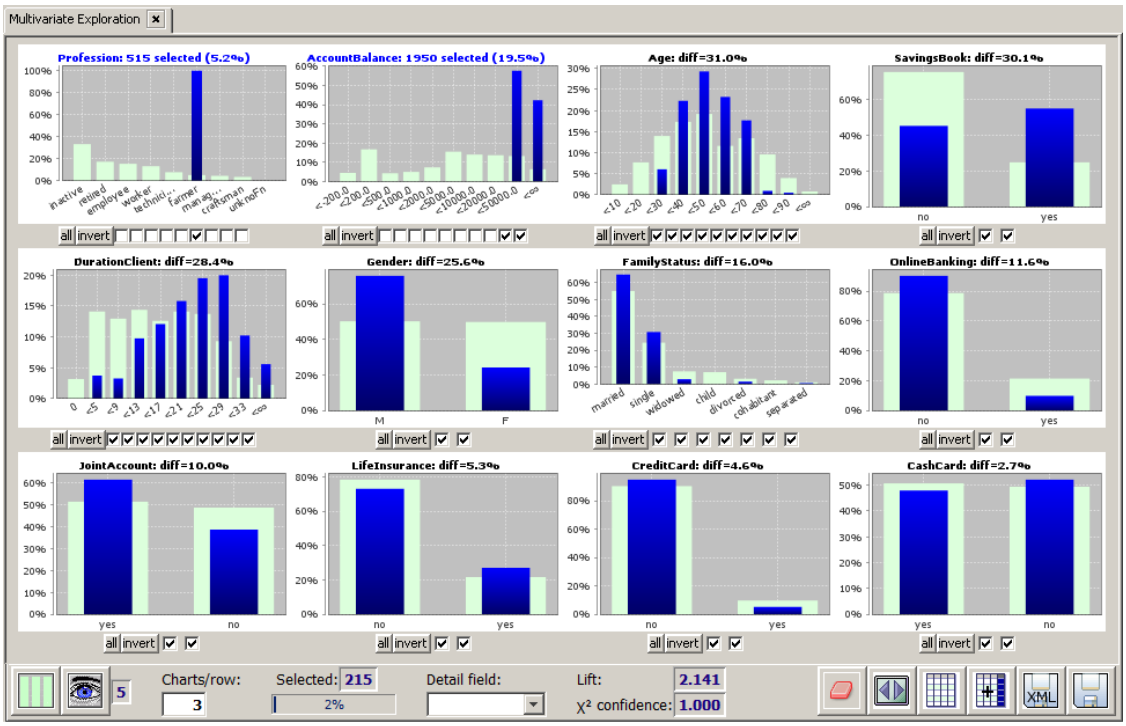
In the following we want to demonstrate some of the options and functions with the help of a concrete example. We again start with the sample data `doc/sample_data/-customers.txt` and we select the 1950 customers with an account balance of at least

20,000. Now we open the pop-up window **Visible fields** and choose **Sort by → rel. difference**.

The displayed field order on the main panel has changed. The selection field `AccountBalance` has been placed first, followed by `Age` and `Profession`, on which the difference between the selected data and the overall data is largest (26.6% and 23.0%).

It is not a big surprise thal pensioners and people elderly people often have a large account balance. More surprising is the fact that farmers are strongly over-represented in the group of people with an accoutn balance of 20000 or more. We want to introspect this group a bit closer, hence we select `Profession=farmer`. Then we again sort the fields by decreasing relative difference. The following picture arises. It shows that farmers with large bank accounts typically have the following characteristics:

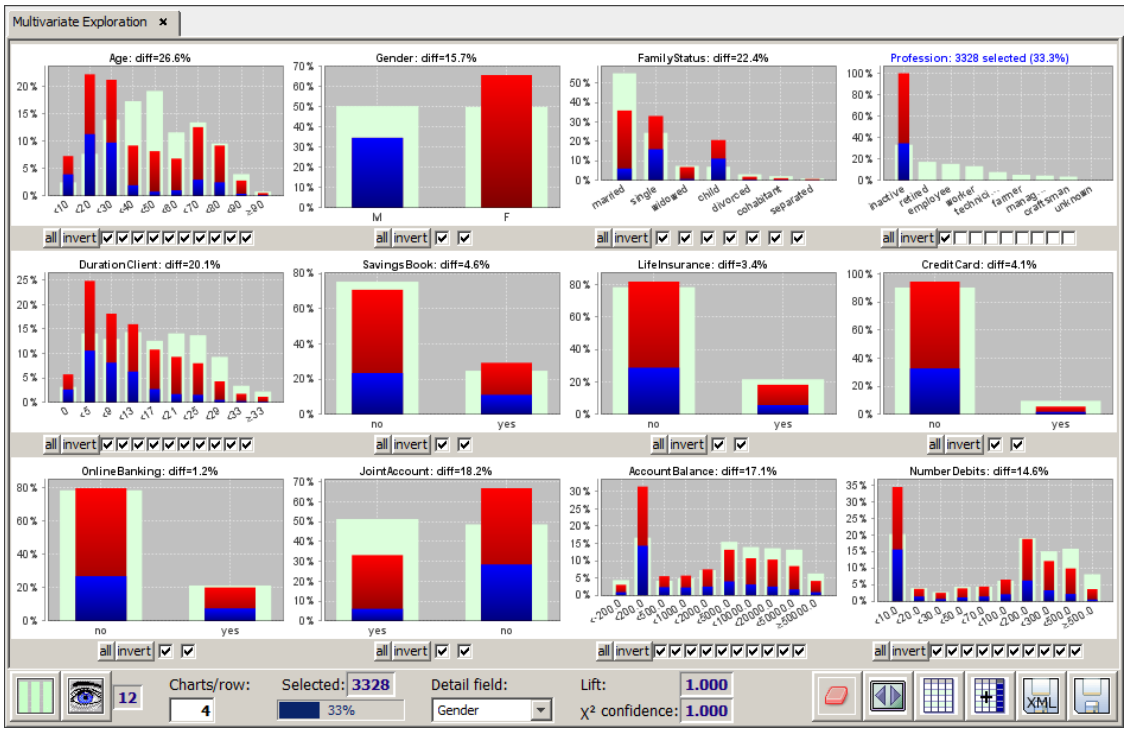
1. A medium age (30 to 70 years)
2. They own a savings book
3. High customer loyalty and long-term customer relationship of more than 20 years
4. They are mainly male.n



3.5.7 Working with detail structure fields

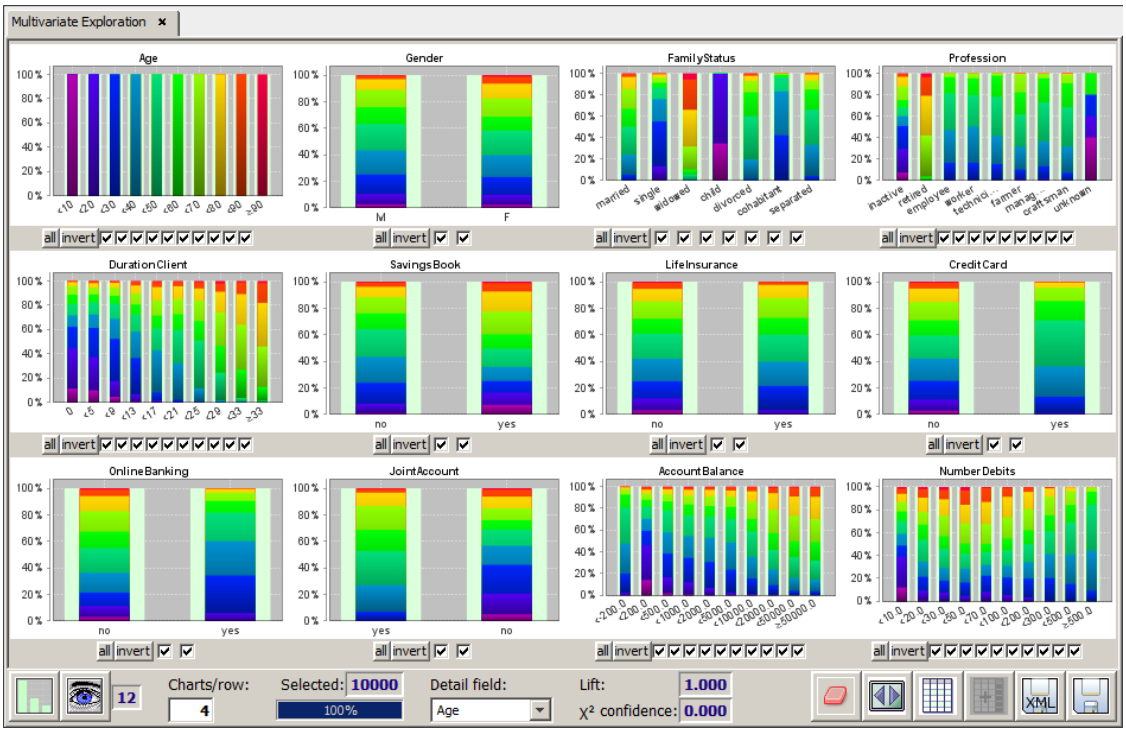
Using the selector box **Detail field** one can add graphical detail information to the histogram bars which represent the selected data subset.

In the screenshot below, for example, the profession `inactive` has been selected, and the gender distribution (male/female) of this selected data subset has been shown on 12 data fields. This was achieved by choosing the field `Gender` as the detail structure field. As it can be seen from the histogram for the data field `Gender`, the red parts of the bars represent the females, the blue ones the males.



We want to mention a particular application scenario of working with a detail field: if all data are selected and if the display mode 'all histogram bars have the same length (100%)' has been selected, specifying a detail field has the effect of creating a collection of many bivariate field-field matrix charts, the y-axis field of all bivariate charts being the detail structure field.

In the following picture we show an example for this application scenario. In the example, the field `Age` has been selected as the detail structure field.

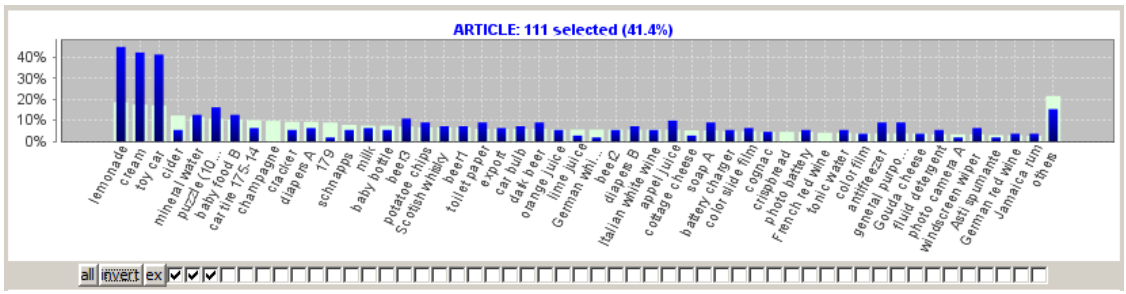


3.5.8 Working with set-valued data fields

If the examined data contain set-valued textual fields, multivariate exploration requires particular care and attention when interpreting the displayed results. Set-valued fields can emerge when a group field has been defined on the data. 'Set-valued' means that within one single data group the field can assume more than one different value. For example, the field `PURCHASED_ARTICLE` could comprise several different purchased articles on the data group `TICKET_ID=3126`.

In the following we want to demonstrate the arising subtleties using the sample data `doc/sample_data/RETAIL_PURCHASES.txt`. We assume that these data have been imported into Synop Analyzer as described in Name mappings, that means with `PURCHASE_ID` as group field and with `doc/sample_data/RETAIL_NAMES_DE_EN.txt` as article names. In these data, the field `ARTICLE` is set-valued with respect to the group field `PURCHASE_ID`: normally, a purchase comprises several different articles.

The screenshot below was obtained by deactivating the three most frequent values in the field `ARTICLE` and by pressing the button **invert** afterwards. We expect to obtain blue bars only for the first three bars in the histogram, but we find blue bars for almost all other values, too. Why?



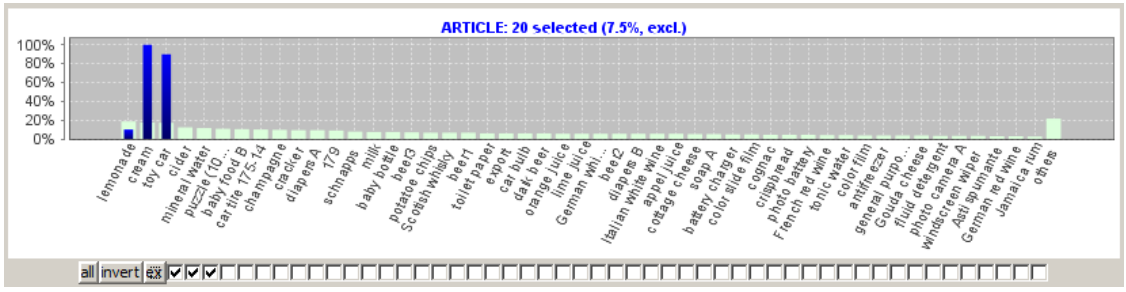
In order to answer this question, we must remember that for the set-valued data field `ARTICLE`, selecting the three articles can have two different meanings:

1. Select all purchases (ticket IDs) which *exclusively* consist of the three selected articles and which do not contain any other article. We call this the *exclusive* selection mode.

2. Select all purchases which contain *at least* one of the selected articles. We call this selection mode the *non-exclusive* mode.n

Obviously, the histogram shown above interprets the selection task as 'non-exclusive' selection. Therefore, the selected purchases also contain many other articles in addition to the three selected articles. Can one switch to the exclusive selection mode in Synop Analyzer? To this purpose there is an additional button **ex** next to the **invert** button. Each click on this button toggles the exclusivity mode of the current selection.

If one clicks on the **ex** button in the histogram shown above, one obtains the following histogram:



As desired, this histogram displays only those 20 purchases in which no other articles than the selected three articles were purchased. The fact that we now are in 'exclusive' mode is indicated by the text **(excl.)** in the title of the histogram chart.

In addition, the software applies the following principles when dealing with set-valued fields:

1. If one starts with a data field without range limitations (all checkboxes marked) and begins deactivating single checkboxes, then Synop Analyzer automatically switches into the exclusive selection mode. This corresponds to the intuitively expected behaviour: by deactivating a checkbox I tell the software that I do not want to see those purchases which contain the deselected article.

2. Pressing the **invert** button does not only invert the selection but also switches from exclusive to non-exclusive mode and vice versa. This is also the intuitively expected behaviour: if I deactivate a value, I want to see the data groups which do not contain the value. If I invert this task then I want to see the data groups which definitely contain the value, but which can contain other values, too). The two selections between which the **invert** switches back and forth are disjunct, and their combination is the entire data.

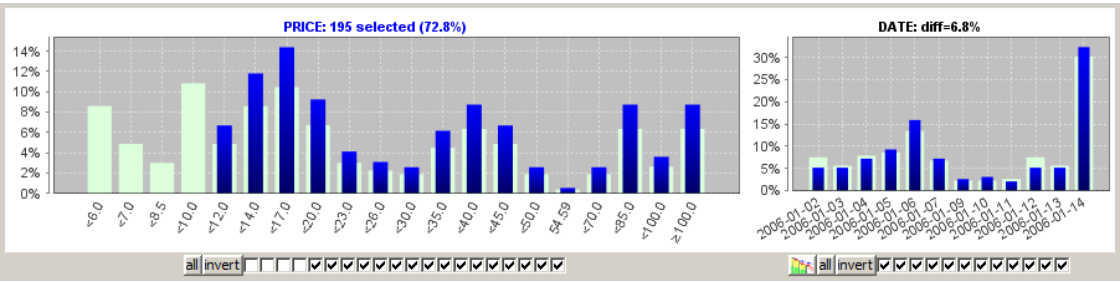
3. All other actions which can be performed using the checkboxes or in the details pop-up view do not change the selection mode.n

3.5.9 Creating forecasts and what-if scenarios

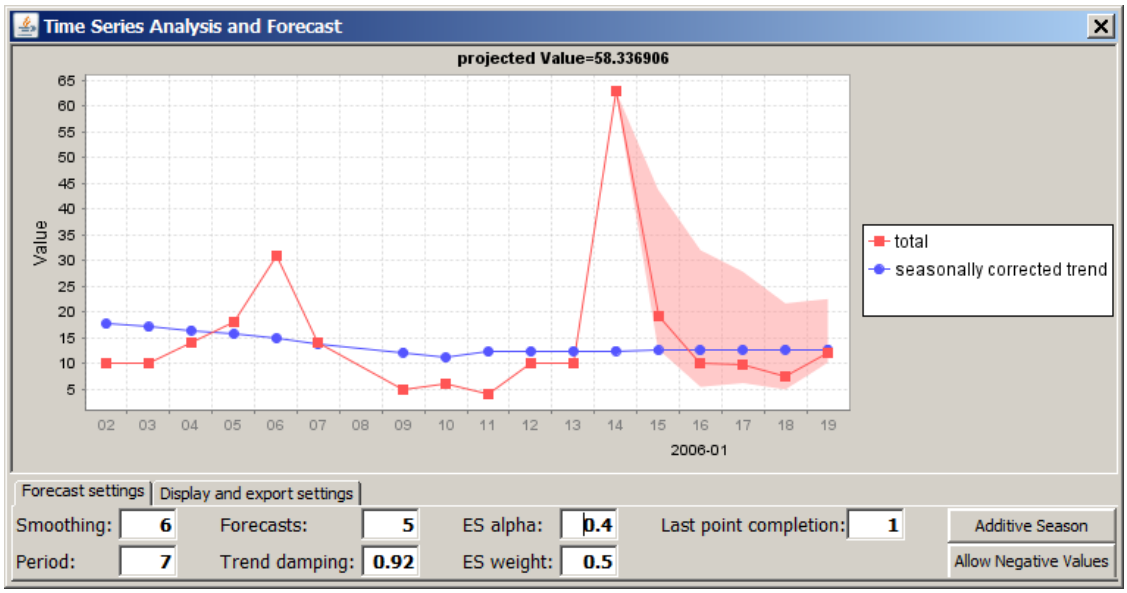
For numeric data fields with a date or timestamp data format, Synop Analyzer is able to start a time series analysis and forecast by clicking on a special button below the field's histogram chart. This special button is situated next to the button **all** and displays a time series plot as button icon.

In the following we want to give an example based on the sample data `doc/sample_data/RETAIL_PURCHASES.txt`. We assume that these data have been imported into Synop Analyzer as described in Name mappings, that means with `PURCHASE_ID` as group field and with `DATE` as order (timestamp) field.

The field `DATE` has the additional time series forecast button. We select all purchase prices of 10 EUR or more, then we press the forecast button.



A pop-up dialog opens up. Its content corresponds to the content of the time series forecast panel which can be started from the analysis module button list in the lower part of the left screen column of Synop Analyzer. We refer to section Time Series Analysis and Forecast for a more detailed description of the available buttons and functions. Here, we only show and example. It shows the forecast created when assuming a period (cyclic pattern) of 7 days and a forecast period (in days) of 5:


Time Series Analysis and Forecast
projected Value=58.336906
Value
65
60
55
50
45
40
35
30
25
20
15
10
5
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
2006-01
total
seasonally corrected trend
Forecast settings
Display and export settings
Smoothing: 6
Forecasts: 5
ES alpha: 0.4
Last point completion: 1
Additive Season
Period: 7
Trend damping: 0.92
ES weight: 0.5
Allow Negative Values

3.6 The Module 'Split Analysis'

3.6.1 Purpose and short description

Split Analysis is a data analysis approach in which two data subsets are selected: a 'test' data set and a 'control' data set. In many use cases, the test data set comprises a data subset whose data records have a certain property in common, for example all men, all customers below the age of 30, all vehicles produced after an improvement measure has been effectuated, etc. The first goal of the analysis is to select a suitable control group which is representative for the test group in all attributes except the ones used for defining the test group. The second goal is to find and quantify significant differences between the test data subset and the control data subset.

3.6.2 Understanding the main panel

The main part of the Split Analysis panel consists of one histogram chart per active data field. Each histogram chart compares a field's value distribution on the currently selected test data (blue bars) to the field's value distribution on the currently control data (red bars) and on the entire data (light green bars).

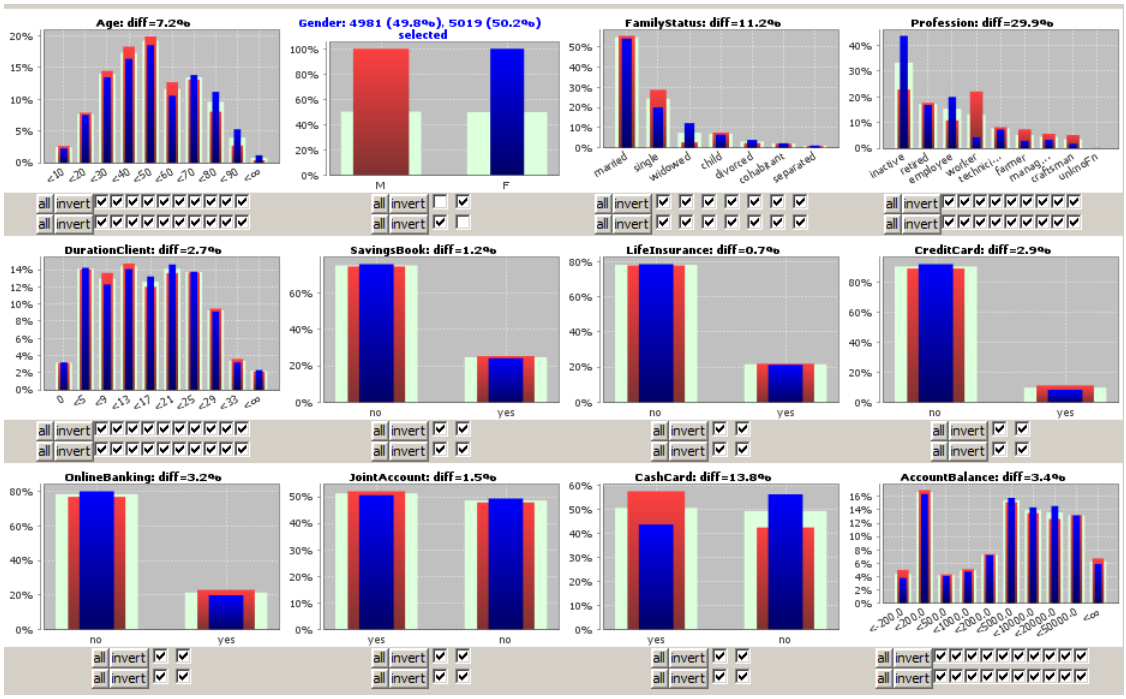
Histograms with more than 36 bars cover the entire screen width, histograms with not more than 18 bars are grouped into tupels of N charts per screen row, where N is the number entered into the tool bar input field named **Charts/row**. If this input field contains the value 0, the software decides autonomously how many charts to put into one screen row. Charts with 19 to 36 bars occupy twice as much horizontal space as the charts with not more than 20 bars. In order to avoid ugly gaps in the arrangements of the charts on screen, the 'large' charts (those with more than 18 bars) are placed before the 'small' charts, that means those with less than 19 bars.

In the histogram charts for non-numeric data fields, the values are arranged by descending occurrence frequency from left to right.If a data field has more then N different values, where N is the number in the input field **#values (text fields)** in the Input Data panel, then only the N most frequent values have been separately recorded when the data were imported. All other values have been summarized into the 'rest' value 'others'. This rest value will be represented in the chart by one single bar with label 'others'. If there is no such 'rest' value in the data, it can still be the case that there are so many different values that it is impossible to draw a histogram bar for each of them. In this case, the histogram chart will be truncated after 80 bars (you can change that value of 80 in the pop-up dialog **Preferences → Multivariate Preferences**). The fact that some bars could not be displayed is indicated by an additional label saying "... ?? others", where ?? is the number of suppressed bars.

Numeric data fields - such as the field `Age` in the picture below - often have so many different values that a binning into a small number of value ranges or intervals is reasonable. The number of bins and the bin boundaries have been defined and can be modified in the Input Data Panel.

By clicking on one of the checkboxes which are situated below each chart, a value selection (restriction) can be defined for the corresponding data field. The upper row of checkboxes specifies the selection defining the test data subset the lower row specifies the selection defining the control data subset.

In the following screenshot, the sample data `doc/sample_data/customers.txt` have been imported into Synop Analyzer. Then, the Split Analysis module has been started and the left checkbox below the chart for the field `Gender` has been deselected for the test data, the right one for the control data. That means, we have defined the female customers as test data subset and the male customers and the control data subset.



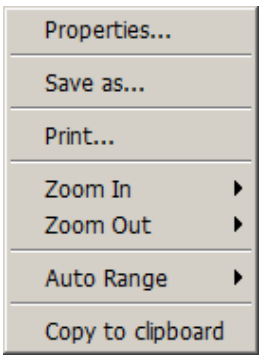
We derive from the picture that the professions of the female customers strongly differ from those of the male customers - more women are employees or inactive whereas much more men are workers - while there is almost no difference between both groups as to the possession rate of savings books or credit cards.

The user can now interactively select an deselect values and value ranges in one or more arbitrary other data fields, independently for the test data and the control data, thereby defining two multivariate data selections. The calculation of the two overall selections is performed on an in-memory representation of the data which is optimized for those multivariate 'slicing' operations over several fields. Therefore, the results can be calculated and displayed within fractions of a second even on multi-gigabyte data.

By drawing with the mouse (keep the left mouse button pressed while moving) on a histogram chart you mark a rectangular region in which you want to zoom in.

By right-clicking on a histogram chart you open the pop-up dialog shown below. In this dialog, you can modify the appearance of the histogram chart (text fonts and sizes, axis

styles, labels, etc.) via the menu item **Properties**. You can also save the chart as PNG graphics, print it or copy it as png graphics object to the system clipboard.

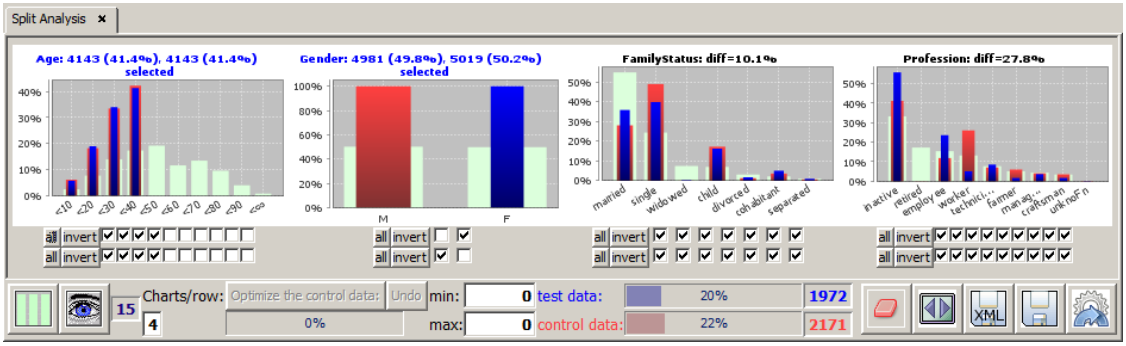


Using the button **Visible fields** in the bottom toolbar, you can hide and remove certain fields from the charts panel in order to get a clearly arranged picture on data with many data fields. In the picture shown at the beginning of this section we have hidden the two fields `NumberCredits` and `NumberDebits`.

3.6.3 Working with the range selector buttons

Now we want to study the possibilites of selecting and deselecting value ranges by means of the button bars below the histogram charts in more detail. To that purpose we focus on a part of the screenshot shown above, namely the histograms and button bars for the four data fields `Age`, `Gender`, `FamilyStatus` and `Profession`.

In addition to the existing range limitation on the field `Gender` we want to restrict the values of the field `Age`, namely we want to focus on the customers below 40 years. To that purpose we could deselect the six rightmost checkboxes under the histogram for field `Age`. A bit faster is the alternative approach of deselecting the four leftmost checkboxes and then clicking on the **invert** button. The **invert** button inverts the existing range selection on a data field. The button **all**removes all ranges restrictions from the field. We perform the value range selection twice: once for the upper, blue data, once for the lower, red data.



The new selection defines 4143 customers in the selected `Age` region. As the intersection with the existing preselection of 4981 female respectively 5019 male customers we get 1972 or about 20% young female and 2171 or about 22% young male customers (these numbers are displayed in and next to the progress bars in the bottom tool bar; the blue bar represents the test data, the red one the control data).

The range restriction in the field `Age` instantaneously changes the heights of the blue and red bars in all other data fields. As expected, the percentage of children and singles in the field `FamilyStatus` have grown significantly. The difference between the two the selected subsets and the light green background distribution on the entire data has grown strongly on most data fields. In contrast, the differences between the two selected groups on the fields `FamilyStatus` and `Profession`, which are displayed in the respective chart titles, have declined. The displayed '**diff** value is calculated as the total length of all parts of the blue bares which exceed the red bars divided by the total length of all blue bars (the latter is always 100% if the respective field is not set-valued).

The chart titles of the fields in which we have specified a range restriction (selection) are displayed in blue; the titles of the 'response' fields in which the observed differences between blue, red and light-green bars are a reaction of range selections in other fields are displayed in black.

3.6.4 Working with detail pop-up dialogs für single fields

A left mouse-click on one of the histogram charts opens a tabular detail statistics which shows the field's values or value ranges and their actual and expected occurrence frequencies on the test (**#test**) and the control data (**#control**). **#expected(test)** is the expected number of test data records under the assumption that the value's relative frequency on the test data is identical to the value's relative frequency on the control data. The columns **difference** and **rel. difference** contain the absolute and relative difference between the actual and the exected occurrence frequency on the test data. Finally, the column **significance** displays the result of a χ^2 significance test which indicates whether the observed difference between actual and expected occurrence frequencies on the test data are statistically significant (significance values close to 1) or not (significance values below 0.95...0.9).

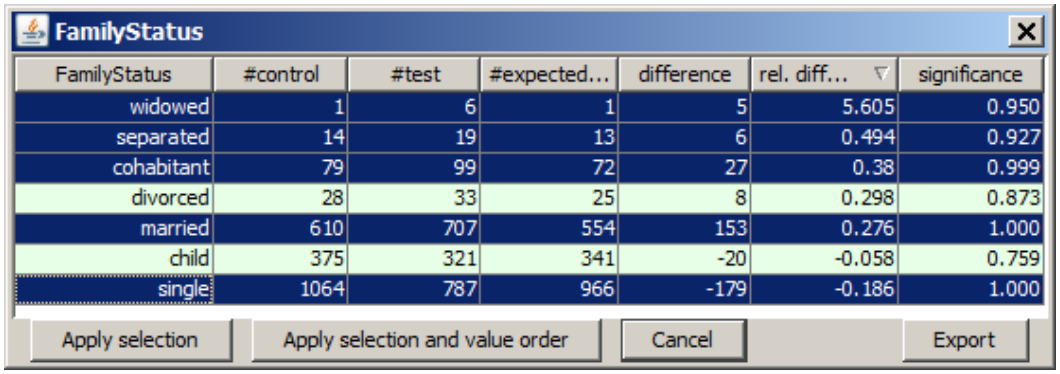
FamilyStatus

FamilyStatus	#control	#test	#expected...	difference	rel. differe...	significance
married	610	707	554	153	0.276	1.000
single	1064	787	966	-179	-0.186	1.000
widowed	1	6	1	5	5.605	0.950
child	375	321	341	-20	-0.058	0.759
divorced	28	33	25	8	0.298	0.873
cohabitant	79	99	72	27	0.38	0.999
separated	14	19	13	6	0.494	0.927

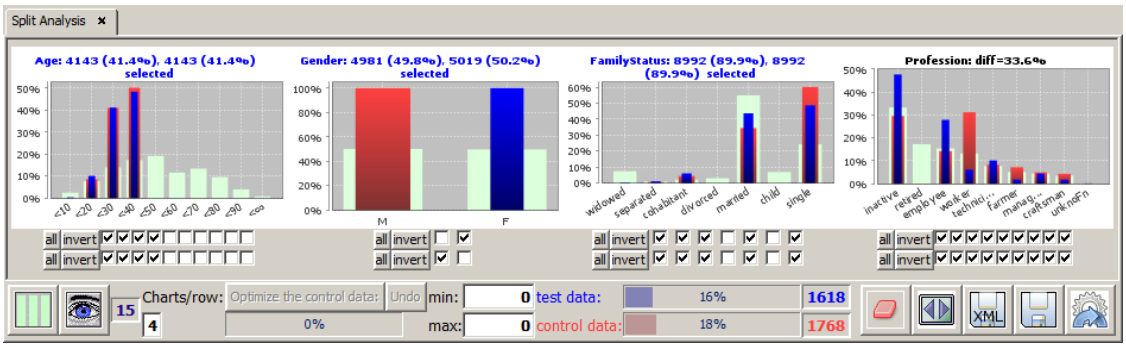
Apply selection | Apply selection and value order | Cancel | Export

If a non-numeric data field has many different values, for example far more than 100, then the available space in the histogram is not sufficient for displaying a separate bar and checkbox for each of them. In this case, the pop-up detail view is the only possibility for seeing all different values and for selecting or deselecting single values which do not figure among the 80 most frequent values. This selection or deselection can be performed by mouse-clicks on certain table rows in the detail view. If you keep the <CTRL> key pressed while clicking, you can select more than one row, by keeping the <SHIFT> key pressed you can select an entire value range. After selecting the desired table rows you activate your selection and close the pop-up view by pressing the button **Apply selection**. Selections in the pop-up view are always applied on both the test and the control data.

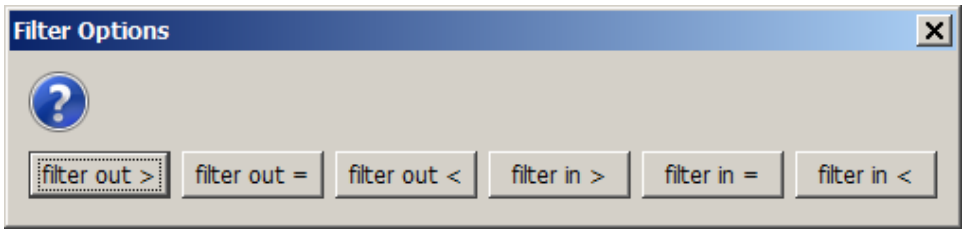
In the details pop-up view you can also reorder the values by pressing on one of the column heads. This sorts the values ascendingly or descendingly by the values of the clicked column. Repeated clicks invert the sorting order. In the screenshot shown below, we have sorted by descending relative difference. This brings the value `widowed` to the top position. Then we have deselected all values on which the differences in relative frequency between the test and the control data is not significant at a confidence level of at least 90%.


FamilyStatus
FamilyStatus | #control | #test | #expected... | difference | rel. diff... | significance
widowed | 1 | 6 | 1 | 5 | 5.605 | 0.950
separated | 14 | 19 | 13 | 6 | 0.494 | 0.927
cohabitant | 79 | 99 | 72 | 27 | 0.38 | 0.999
divorced | 28 | 33 | 25 | 8 | 0.298 | 0.873
married | 610 | 707 | 554 | 153 | 0.276 | 1.000
child | 375 | 321 | 341 | -20 | -0.058 | 0.759
single | 1064 | 787 | 966 | -179 | -0.186 | 1.000
Apply selection
Apply selection and value order
Cancel
Export

If we now leave the pop-up window by pressing the button **Apply selection and value order**, both the new value ordering and the value selection is applied to the histogram chart:

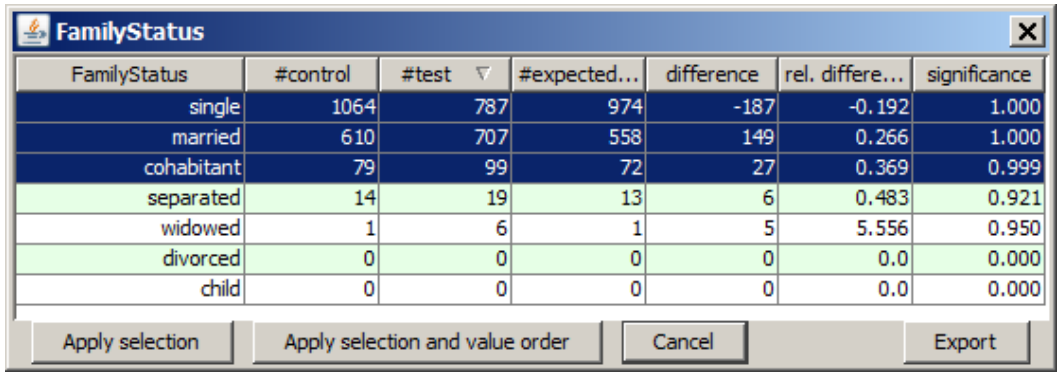


The details pop-up view offers yet another feature: if you right-click on one of the table cells, the following options dialog pops up:



This dialog permits selecting or deselecting all table rows whose values in the column in which the click was performed are in a certain value range, and this selection can be performed by one single click. This is an enormous reduction of effort especially if the field contains hundreds or thousands of different values.

The following picture results from right-clicking on the value `99` in the column `#test` and by choosing the option **deactivate <** in the options dialog. This choice deselects all table rows which have a value of less than `99` in the column `#test`.



3.6.5 The bottom toolbar

The tool bar at the lower screen border provides the following buttons and functions:

- :
  Toggle the histogram display mode in the default display mode, the sum of all light-green background bar heights is 100% ('sum mode'). Pressing this button switches between that default mode a second mode in which each single light-green background bar is rescaled to 100% ('single mode'). This second mode is particularly useful for studying the relative frequency differences between the selected data and the overall data on the various values or value ranges of a data field.

- :
  Via this button you open a pop-up dialog which permits to hide certain data fields from the histogram chart panel. This feature is described in more detail in section Rearranging and suppressing fields. The blue number to the right of the **Visible fields** button shows the total number of remaining visible fields.

- **Charts/row**
  In this input field you can specify how many of the 'normal' histogram charts with not more than 18 bars should be put into one single screen row. The smaller the number, the larger will be each single histogram chart.

- **Optimize the control data**, **Undo**, **min** and **max**:
  Using these buttons you can sample a subset of the control data which is representative for the test data with respect to certain data fields which you have defined in advance. This function will be described in more detail in section Optimize the controll data.

- **Progress bars and adjacent numeric output fields:**
  The progress bars with the labels **Test data** and **Control data**and the adjacent text fields show the size of the currently selected subsets of the data: the number in the progress bars is the percentage of the entire data; the number to the right of the progress bar is the absolute number of selected data records (or data groups if a group field has been specified).

- :
  Undo all range restrictions; select all data records.

- :
  By clicking on this button you re-draw all histogram charts, thereby adapting their size to the current screen width.

-

  :
  By pressing this button, you can save the currently active data import settings and all settings performed in this module to a persistent XML parameter file. This file can later be opened via Synop Analyzer's main menu (**Analysis** → **Run Split Analysis**). In this way you can exactly reproduce the current data analysis screen without to be obliged to re-enter all settings and customizations.

- :
  Export the current data exploration results within this module into a spreadsheet in `.xlsx` format (MS-Excel© 2007+). The spreadsheet contains several worksheets: one with a single PNG graphics for each histogram chart, one with a single PNG graphics for all charts, a data sheet which contains the selected data records, and one more worksheet for each detail pop-up window which ever has been opened by mouse-clicking on one of the histogram charts.

- :
  This button opens a pop-up dialog in which you can define a series of many consecutive split analyses. The first analysis within the series is performed with the currently active parameter settings, in each subsequent analyses, the data split into test and control data is slightly modified in a way specified in the pop-up dialog. When the pop-up-dialog is finished, an XML parameter file and an executable batch file are created. The batch file calls the Synop Analyzer command line processor `sacl` with the XML parameter file as command line argument. This function is described in more detail in section Automatized series of split analyses.

3.6.6 Rearranging and suppressing fields

Clicking on the button **Visible fields** opens a pop-up dialog in which the following actions can be performed:

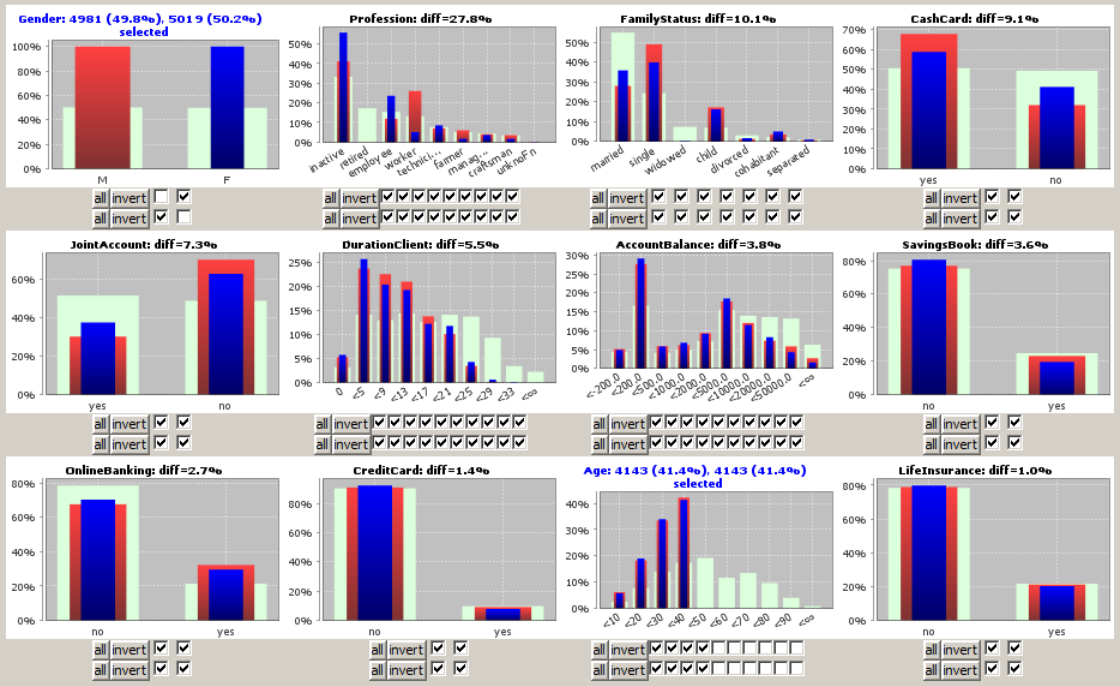
- Hide certain data fields so that no histogram is displayed for them and they are ignored in later control data optimization steps. You can hide a field by left-clicking the field name while keeping the <CTRL> key pressed.

- Rearrange the histograms on screen: if you draw a field name with the mouse to another vertical position and release the left mouse key there, the field name is moved to the new position. Note: moving a field name is only possible within its 'group'. The data fields with many different values and large histogram charts form the first group, the fields with normally sized charts form the second group.

- Sort the data fields with respect to a user-selected filter criterion. The pull-down menu named **Sort by** at the lower border of the **Visible fields** pop-up dialog makes it possible to sort and reorder the displayed histograms with respect to a couple of sorting criteria. The meaning of the criteria **lexical field order** and **field order in the data** should be evident. The criterion **rel. difference** sorts the fields on which a manual range restriction has been defined at first place and then the other fields sorted by descending `diff` value. The criterion χ^2 **conf** also places the fields with manual range restrictions in front, followed by the other fields sorted by decreasing χ^2 confidence value. The χ^2 confidence value indicates the level of confidence of the assertion that the value distribution of the blue test data significantly differs from the value distribution of the red control data. In general, this criterion has some similarity and correlation with the criterion **rel. difference**. However, a small relative difference of, say, 1% can be very significant on a field with many data records only very few different field values, whereas a relative difference of 10% can be non-significant on a field with many different values and few data records.

- Exchange the quantitative difference measure shown in the charts' titles. After selecting the option **Sort by** → χ^2 conf</but>, in the **Visible fields** dialog, the chart titles display the difference measure χ^2 `conf`. Sorting by **rel. differeence** switches back to displaying the relative difference (`diff`).

In the following we want to demonstrate some of the options and functions with the help of concrete examples. We again start with the sample data `doc/sample_data/-customers.txt` and with the selection discussed in the previous section: female customers below 40 years as test group, male customers below 40 as control group. Now we open the pop-up dialog **Visible fields** and hide the two fields `NumberCredits` and `NumberDebits` by left-clicking the two field names while keeping the <CTRL> key pressed. Then we choose **Sort by → rel. difference**.



The field order and the number of displayed fields in the main panel changes: the field `Gender`, in which the two selected groups have a relative difference of 100%, is placed at the top position, followed by the fields `Profession` and `FamilyStatus` on which the difference between young males and femals is strongest (27.8% respectively 10.1%).

3.6.7 Working with set-valued data fields

If the examined data contain set-valued textual fields, the split analysis requires particular care and attention when interpreting the displayed results. Set-valued fields can emerge when a group field has been defined on the data. 'Set-valued' means that within one single data group the field can assume more than one different value. For example, the field `PURCHASED_ARTICLE` could comprise several different purchased articles on the data group `TICKET_ID=3126`.

The difficulties when dealing with set-valued fields is caused by the fact that it is not any more unambiguously clear what activating or deactivating a check box representing a histogram bar means:

1. Select those data groups which *only* have the selected values but no other values. We call this mode the *exclusive* mode.

2. Select those data groups on which the selected values are present *among others*. We call this mode the *non-exclusive* mode.n

In the reference documentation of the module Multivariate Exploration we show in detail how Synop Analyzer can switch between these two different selection modes. That explanation applies one to one also to the split analysis module, therefore we refer to that part of the documentation and do not repeat the explanations here.

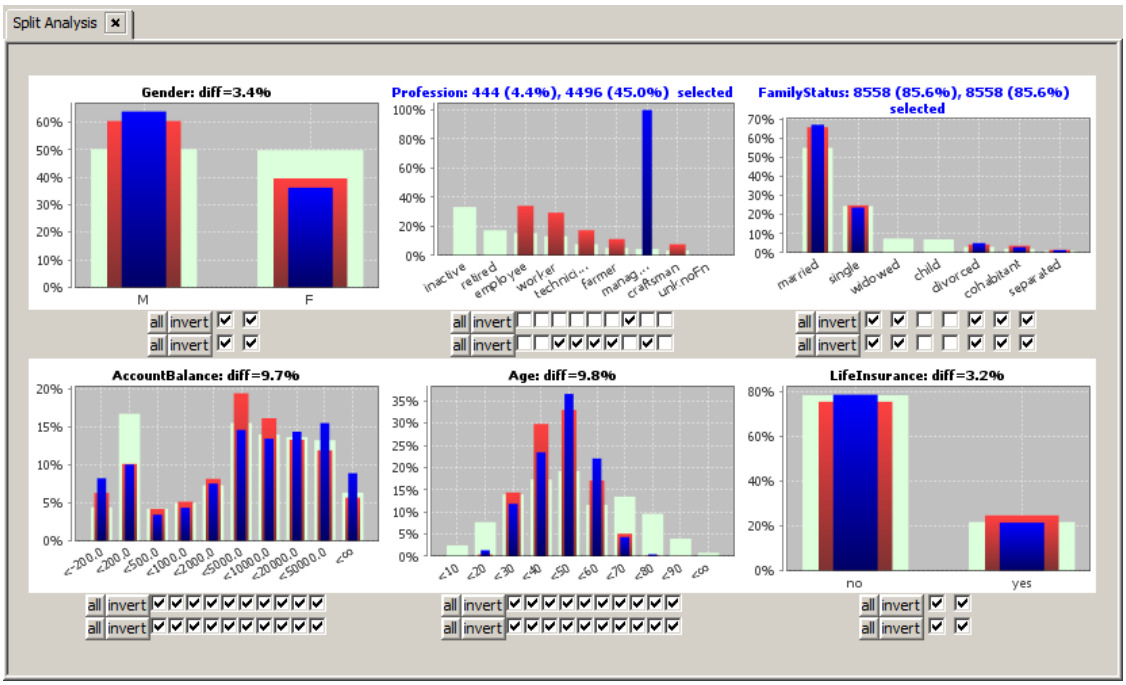
3.6.8 Optimizing the control data

A split analysis is performed with the aim of finding significant differences in the value distributions of one or more 'target' data fields between two data subsets: the 'test' subset, whose values have certain values in one or more 'selector' data fields, and the 'control' subset, whose valuesdo not have those values in the selector data fields. Unfortunately, in most real-world situations, there are inevitably many other differences between the two data subsets in addition to the desired ones. Therefore, one can not be sure whether the observed differences in the 'target' fields are caused by the controllable differences in the 'selector' fields or whether they are due to uncontrollable differences in some other data fields.

In order to make this more concrete, let us consider an example from applied social studies based on the sample data `doc/sample_data/customers.txt`. Using these data, we want to quantitatively verify or falsify the following hypothesis:

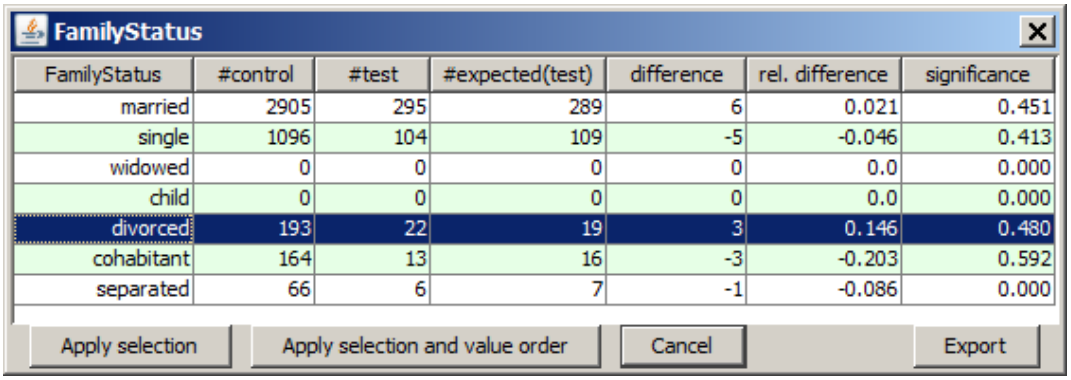
»the Managers are more frequently divorced than people with other professions but similar socio-economic background.«

The available data contain six data fields which define the profession, the marital status and the socio-economic background: `Gender`, `FamilyStatus`, `Profession`, `Age`, and the 'wealth-indicators' `LifeInsurance` and `AccountBalance`. We want to verify the hypothesis stated above by selecting a suitable group of managers as the test group and a group on non-managers as control group.



We import the data and start the module **Split Analysis**. In this module, we use the pop-up dialog **Visible fields** for hiding all data fields but the six fields listed above. In the histogram of the field `FamilyStatus` we deselect (for both the test and the control group) the values which do not match with professionally active persons: `widowed` and `child`. In the field `Profession`, we select the value `Manager` as test group and all other professions except the values `inactive`, `Pensioner` and `unknown` as the control group.

When we open the details view for the field `FamilyStatus` by left-clicking on the histogram chart, our hypothesis seems to be proved - at least by trend.



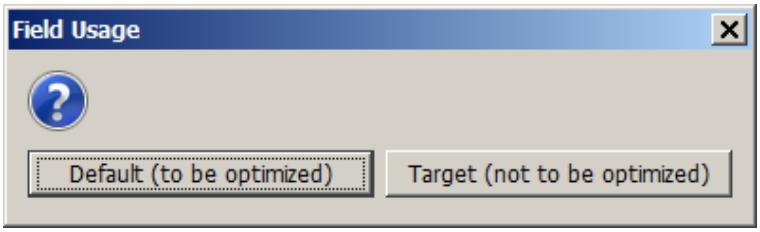
FamilyStatus	#control	#test	#expected(test)	difference	rel. difference	significance
married	2905	295	289	6	0.021	0.451
single	1096	104	109	-5	-0.046	0.413
widowed	0	0	0	0	0.0	0.000
child	0	0	0	0	0.0	0.000
divorced	193	22	19	3	0.146	0.480
cohabitant	164	13	16	-3	-0.203	0.592
separated	66	6	7	-1	-0.086	0.000

The table row highlighted in blue contains the result we are interested in. The row reads as follows: In the test data (managers) there were 22 divorced persons. If the percentage of divorced persons was identical to the percentage of divorced persons in the control group, we would only have 19 divorced managers. 22 minus 19 is an absolute difference of 3 and a relative difference of +14.9%. Unfortunately, the data sample (the number of

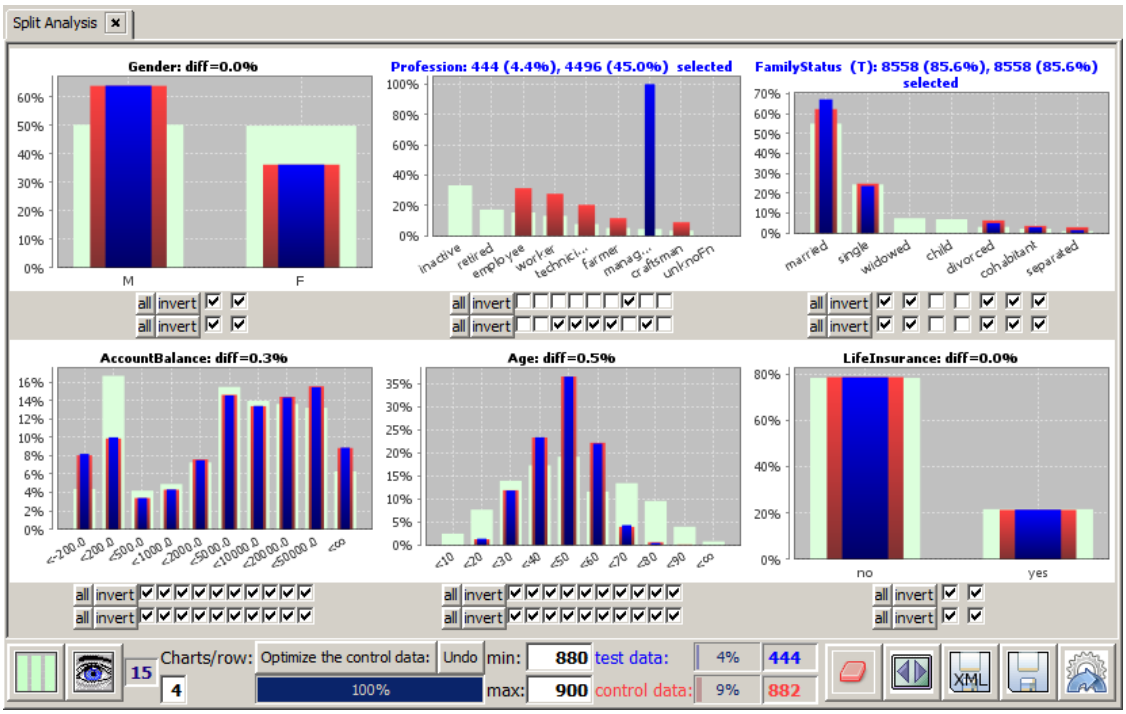
cases) is not large enough so that the result is not yet really significant (confidence level strongly below 90%).

However, the preliminary result stated above is not really valid. The control group differs significantly from the test group in the value distributions of the data fields `Age`, `Gender`, `LifeInsurance` and `AccountBalance`. Therefore, it is unclear whether the observed differences in divorce rates are caused by the differring professions or the differences in the other fields.

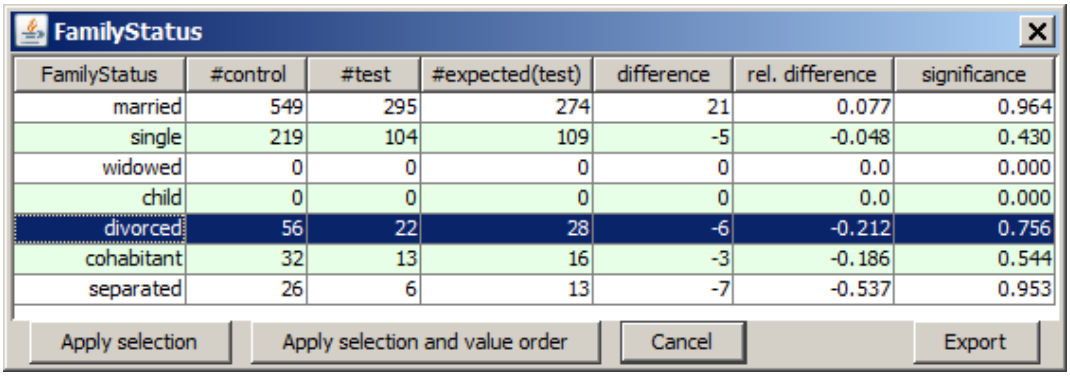
Here, we can use Synop Analyzer's control data optimization feature, which aims at making the control data 'representative' for the test data in a couple of user-defined data fields. First, we have to tell Synop Analyzer which is the target field of our hypothesis. To that purpose, we open the **Visible fields** dialog and right-click on the field name `FamilyStatus`. A new pop-up dialog appears in which we select the option **Target field (distribution will not be optimized)**. After closing the window **Visible fields** the histogram of the data field `FamilyStatus` carries an additional **(T)** (for 'target') in its chart title.



Now we optimize the control data, making them representative for the test data in all data fields but the target field and the selector field `Profession`. We use the tool bar fields **min:** and **max:** to tell the software how large the new controll data should be. The size of the test data is 440 records. We think that a size of the control data of about twice the size of the test data should be enough, therefore we enter `880` as the minimum and `900` as the maximum value. Then we press the button **Optimize the control data**.



A moment later, the control data size has dropped to 882 data records, and the control data's value distributions on the four data fields to be optimized are perfectly identical with the respective value distributions of the test data. If we now open the details view of the field `FamilyStatus`, we get a result which differs strongly from our preliminary result:



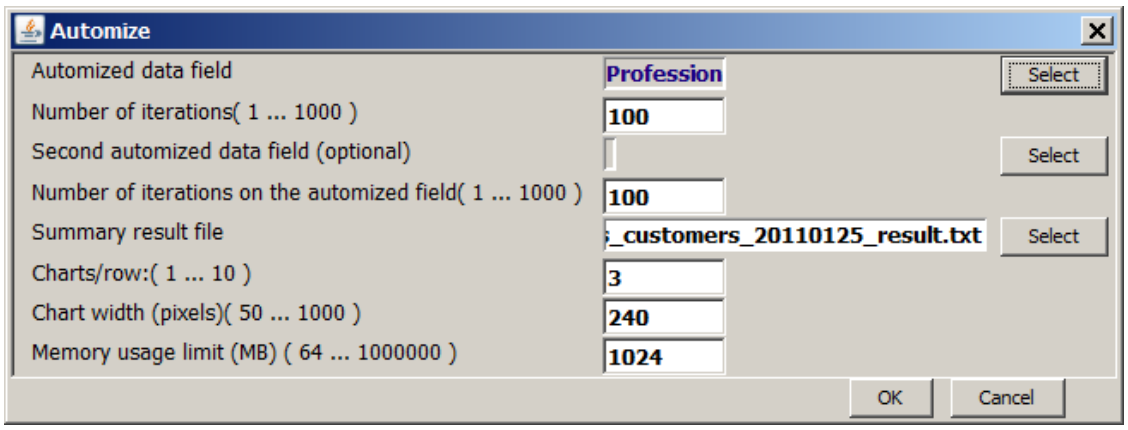
FamilyStatus	#control	#test	#expected(test)	difference	rel. difference	significance
married	549	295	274	21	0.077	0.964
single	219	104	109	-5	-0.048	0.430
widowed	0	0	0	0	0.0	0.000
child	0	0	0	0	0.0	0.000
divorced	56	22	28	-6	-0.212	0.756
cohabitant	32	13	16	-3	-0.186	0.544
separated	26	6	13	-7	-0.537	0.953

We see that when working with 'representative' control data, the profession `Manager` has no pushing impact on the divorce rate. On the contrary, there are less divorced managers then expected from the other profession groups (even though this tendency is not really statistically significant, the confidence level is only 75%). We understand how important it is to optimize the control data before deducing conclusions from a split analysis.

3.6.9 Automatized series of split analyses

Often, it is desirable to perform large series of similar split analyses. For example, we could repeat the split analysis performed in the previous section for all other professions, not only for managers. And maybe we would like to repeat the entire series of split analyses every 3 months in order to monitor socio-demographic trends.

For both goals, an automatized scheduling of many similar split analysis tests is required. Synop Analyzer provides the button **Automatize** for that purpose. The button creates an executable batch file in which the command line processor sacl is called with a suitable command line argument in order to perform the entire series of tests without any user interaction. Pressing the button **Automatize** first opens a file selection dialog in which one can define the file name of the batch file to be created. Then the following dialog opens up:



In this view we define in the first row, over which data field the series of split analysis tasks is supposed to iterate. The selection box offers all data fields in which exactly ine field value is currently activated on the test data and some other values are activated on the control data. In our example, only the field `Profession` satisfies these requirements.

The second row defines the maximum number of iterations over the field specified in the first row the series is to be terminated. The default value is 100. Since we only have 6 different professions in the data, we can leave that value unchanged, it has no effect; we could also enter `6` here.

In the third and fourth row, one can define a second data field to iterate over. In our example, there is no suitable second field for iterating over.

Then, we specify the name of the summary result file - a <TAB> separated text file which can be opened in MS Excel and which contains one line of summary information for each single split analysis performed during the series. Finally, there are three parameters with which you can modify the graphical representation of the single tests' results, and a parameter which defines the maximum amount of computer memory to be available when running the automatized analysis series.

In addition to the summary result file, the automatized series of tests will create one separate spreadsheet file per single test (iteration) which contains the same results that one would obtain if one manually executed the singe split analysis and then pressed the **Export** button in the bottom tool bar of the split analysis panel.

As soon as one presses **OK** in the pop-up window, the batch file is generated can be started any time.

3.7 The Time Series Analysis and Forecasting module

3.7.1 Purpose and short description

In the Time Series panel, time series can be explored and forecasts can be calculated using various forecasting algorithms. This module can only be started on data which fulfill the following requirements:

1. An order field has been defined in the Select active fields dialog. This field will be the x-axis field in the time series charts.
2. A weight/price field has been defined in the Select active fields dialog. This field will be the y-axis field in the time series charts.
3. Not more than two further active fields exist (plus optionally a group field). All other fields have been deactivated in the Select active fields dialog.n

3.7.2 Required data properties

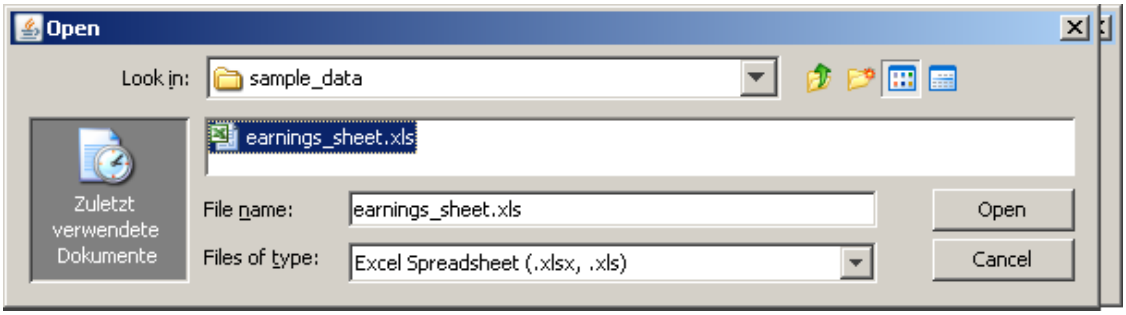
In the following sections we will explain the features and functions of the time series analysis module at the example of the MS Excel file `doc/sample_data/earnings_sheet.xls`. The file contains the monthly earnings sheet for a small company with two locations for the period from January 2006 to March 2009. The figure below shoes a part of this Excel sheet:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**Earnings Sheet 2006 - 2009**															
2		Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 2	Location 2
3		*01.2006*	*02.2006*	*03.2006*	*04.2006*	*05.2006*	*06.2006*	*07.2006*	*08.2006*	*09.2006*	*10,2006*	*11,2006*	*12,2006*	*13,2006*	*01.2006*	*02.2006*
5	**Total Sales**	**1403.6**	**1536.2**	**1117.8**	**981.0**	**874.5**	**1149.3**	**1288.2**	**1365,1**	**1070,9**	**923,0**	**845,8**	**881,3**		**863,7**	**851,9**
7	Subcontracting Cost	151,7	111,4	76,4	52,0	50,4	48,3	67,9	66,9	67,9	24,4	27,6	63,1	30,1	118,1	104,9
8	**Gross Profit I**	**1251,9**	**1424,8**	**1041,4**	**929,0**	**824,1**	**1101,0**	**1220,3**	**1298,2**	**1003,0**	**898,6**	**818,2**	**818,2**	**-30,1**	**745,6**	**747,0**
9	Personnel Cost	710,8	688,0	510,5	569,3	416,0	432,6	440,1	495,1	524,1	392,1	358,5	392,4		383,2	374,9
10	Supplies,Energy	144,4	108,4	85,3	66,9	35,3	32,0	45,0	35,1	60,9	51,7	50,9	117,3	12,5	70,8	23,6
11	Rental & Lease	31,3	27,6	29,8	24,4	153,2	155,6	155,6	150,6	153,1	151,4	153,2	154,9		73,8	73,8
12	Maintenance & Repair	22,2	15,7	15,6	12,9	8,7	10,5	33,3	38,1	17,9	12,5	7,4	29,1		1,2	6,9
13	Insurance,Fee,Taxes	15,2	13,4	14,7	13,4	13,4	13,4	13,4	14,2	13,4	13,4	13,4	13,4	39,3	4,2	4,2
14	Other Operating Charges	7,9	180,2	158,4	61,1	55,2	43,3	43,7	43,3	42,3	43,0	42,8	57,4	-44,2	4,0	3,5
15	Depreciations of Fixed Asse	177,3	177,3	177,3	177,3	117,1	118,8	118,8	118,8	118,8	118,8	118,8	133,8		43,5	43,5
16	**Gross Profit II**	**142,8**	**214,2**	**49,8**	**3,7**	**25,2**	**294,8**	**370,4**	**403,0**	**72,5**	**115,7**	**73,2**	**-80,1**	**-37,7**	**164,9**	**216,6**
17	Total Indirect Cost	78,8	88,7	48,7	61,9	53,0	41,2	47,7	37,3	45,2	45,3	47,5	15,4	-1,9	44,0	43,3
18	**EBIT**	**64,0**	**125,5**	**1,1**	**-58,2**	**-27,8**	**253,6**	**322,7**	**365,7**	**27,3**	**70,4**	**25,7**	**-95,5**	**-35,8**	**120,9**	**173,3**
19	Financial Income(Charges)	44,3	41,8	49,2	48,5	10,9	8,4	10,5	10,9	12,5	12,5	13,4	13,4	50,2	8,4	11,9
21	**EBT**	**19,7**	**83,7**	**-48,1**	**-106,7**	**-38,7**	**245,2**	**312,2**	**354,8**	**14,8**	**57,9**	**12,3**	**-108,9**	**-86,0**	**112,5**	**161,4**

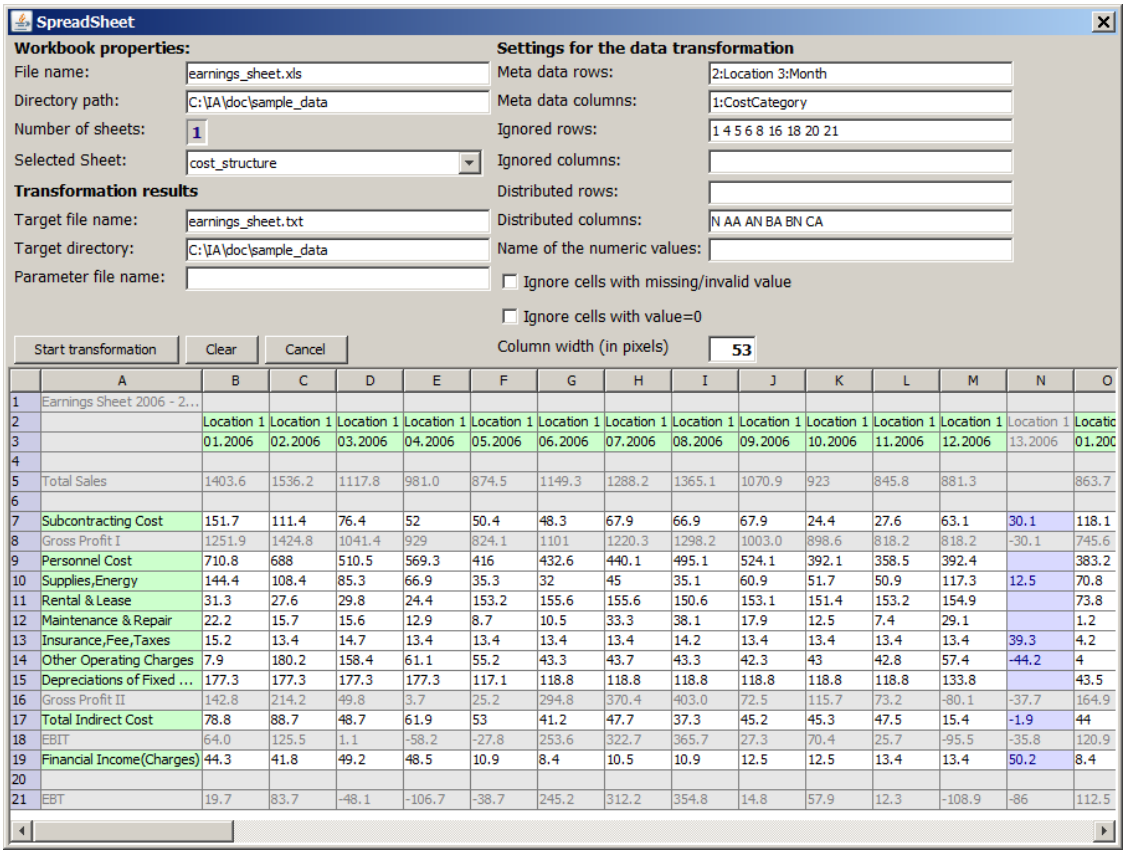
In the present form, the data are not really suitable for being used by a forecasting and trend analysis tool: in the Excel sheet, meta data information (such as location, date or cost category) is intermixed with number cells, empty space cells, formula cells (such as EBT or Gross Profit II) and auxiliary title or text cells. Furthermore, the sheet contains accountant's corrections at year end (such as the column `13.2006` in the picture above); these corrections have to be distributed on the 12 months of the preceding year before the corresponding time series can be used for a forecast or trend analysis. We will see

that Synop Analyzer supports various preprocessing steps on this input sheet in order to overcome the aforementioned problems.

From the Synop Analyzer main menu, we select **File → Import data from spreadsheet**. A file chooser dialog opens up.



We select the file `doc/sample_data/earnings_sheet.xls` in the file chooser dialog. A new **Spreadsheet** window opens on the main canvas:



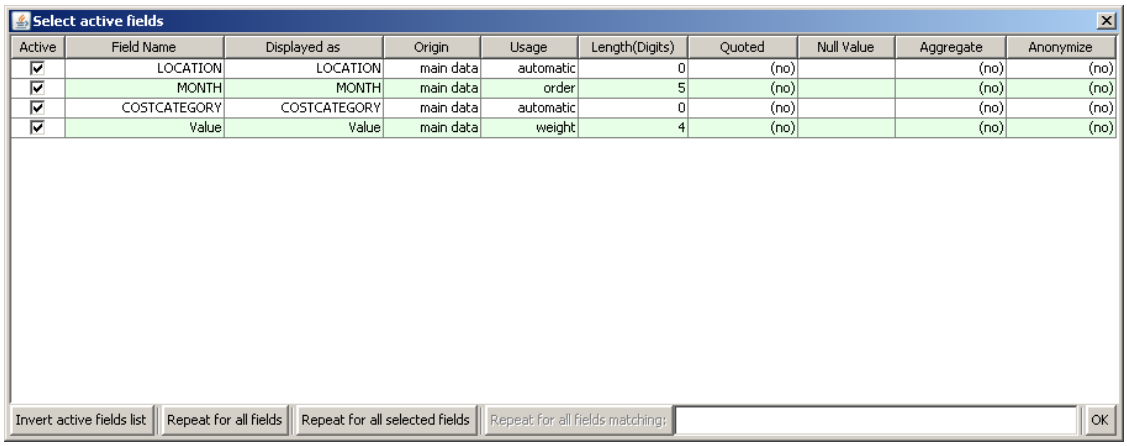
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Earnings Sheet 2006 - 2...														
2		Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Location 1	Locatic
3		01.2006	02.2006	03.2006	04.2006	05.2006	06.2006	07.2006	08.2006	09.2006	10.2006	11.2006	12.2006	13.2006	01.200
4															
5	Total Sales	1403.6	1536.2	1117.8	981.0	874.5	1149.3	1288.2	1365.1	1070.9	923	845.8	881.3		863.7
6															
7	Subcontracting Cost	151.7	111.4	76.4	52	50.4	48.3	67.9	66.9	67.9	24.4	27.6	63.1	30.1	118.1
8	Gross Profit I	1251.9	1424.8	1041.4	929	824.1	1101	1220.3	1298.2	1003.0	898.6	818.2	818.2	-30.1	745.6
9	Personnel Cost	710.8	688	510.5	569.3	416	432.6	440.1	495.1	524.1	392.1	358.5	392.4		383.2
10	Supplies,Energy	144.4	108.4	85.3	66.9	35.3	32	45	35.1	60.9	51.7	50.9	117.3	12.5	70.8
11	Rental & Lease	31.3	27.6	29.8	24.4	153.2	155.6	155.6	150.6	153.1	151.4	153.2	154.9		73.8
12	Maintenance & Repair	22.2	15.7	15.6	12.9	8.7	10.5	33.3	38.1	17.9	12.5	7.4	29.1		1.2
13	Insurance,Fee,Taxes	15.2	13.4	14.7	13.4	13.4	13.4	13.4	14.2	13.4	13.4	13.4	13.4	39.3	4.2
14	Other Operating Charges	7.9	180.2	158.4	61.1	55.2	43.3	43.7	43.3	42.3	43	42.8	57.4	-44.2	4
15	Depreciations of Fixed ...	177.3	177.3	177.3	177.3	117.1	118.8	118.8	118.8	118.8	118.8	118.8	133.8		43.5
16	Gross Profit II	142.8	214.2	49.8	3.7	25.2	294.8	370.4	403.0	72.5	115.7	73.2	-80.1	-37.7	164.9
17	Total Indirect Cost	78.8	88.7	48.7	61.9	53	41.2	47.7	37.3	45.2	45.3	47.5	15.4	-1.9	44
18	EBIT	64.0	125.5	1.1	-58.2	-27.8	253.6	322.7	365.7	27.3	70.4	25.7	-95.5	-35.8	120.9
19	Financial Income(Charges)	44.3	41.8	49.2	48.5	10.9	8.4	10.5	10.9	12.5	12.5	13.4	13.4	50.2	8.4
20															
21	EBT	19.7	83.7	-48.1	-106.7	-38.7	245.2	312.2	354.8	14.8	57.9	12.3	-108.9	-86	112.5

The upper right part of the window contains several input fields in which we can specify how the spreadsheet is to be used. The lower part of the window shows the effects of these specifications.

- In the field **Meta data rows** we specify that the second and third row of the Excel sheet contain two different kinds of meta data information which we would like to use in our analysis. By typing `2:Location 3:Month` we indicate that we want to refer to the meta data in row 2 under the label `Location` and to the meta data in row 3 under the label `Month`.
- Similarly, we indicate that the first column (`A`) of the Excel sheet contains a meta data information to which we want to refer under the label `CostCategory`.
- Our goal in this example is a cost structure analysis. Therefore, we only maintain the rows containing the various cost category figures and we discard the other figures such as `Total Sales`, `Gross Profit`, `EBIT` or `EBT`. That's why we type `1 4 5 6 8 16 18 20 21` into the field **Ignored rows**.
- The columns `N`, `AA`, `AN`, `BA`, `BN` and `CA` of the Excel sheet contain the accountant's corrections at year-end for the two locations. We want to distribute these corrections equally on the 12 months preceding the correction and discard the correction month 13. Therefore, we enter `N AA AN BA BN CA` in the input field **Distributed columns**.

The specifications decribed above are automatically reflected by an adapted coloring scheme in the tabular representation of the currently active spreadsheet in the lower part of the screen: spreadsheet cells containing meta data information are displayed with a green background, cells with values to be distributed among other cells have a blue background, cells which are to be ignored are grayed out and 'normal' value cells are displayed with white background.Finally, we click on the **Start Transformation** button. An instant later, the pop-up window closes and the transformed flat file `earnings_sheet.txt` is written into our chosen target directory.

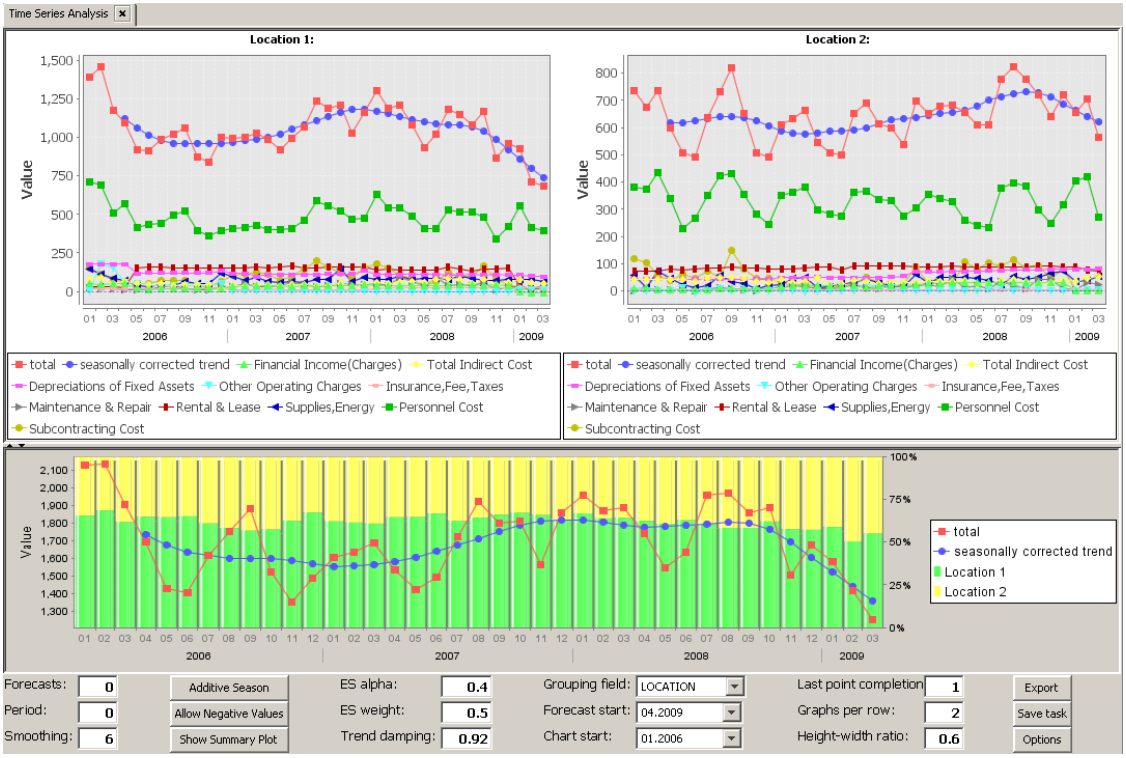
The generated file contains the columns `Location`, `Month`, `CostCategory` and `Value`. The new file is suitable for a statistical analysis using the entire set of Synop Analyzer functions. The time series analysis module requires that an order (timestamp) field and a weight (cost) field have been specified on the input data source.



In the dialog **Select Active Fields** we manually define the field usage of `Month` as **order** and the usage of `Value` as **weight**. Then we press **OK**. After that, we re-read the data by pressing the **Start** button. The button **Time Series Analysis** is now active.

3.7.3 The summary plot

We click on the **Time series analysis** button for starting the forecasting and trending analysis. A new tab **Time Series** pops up:



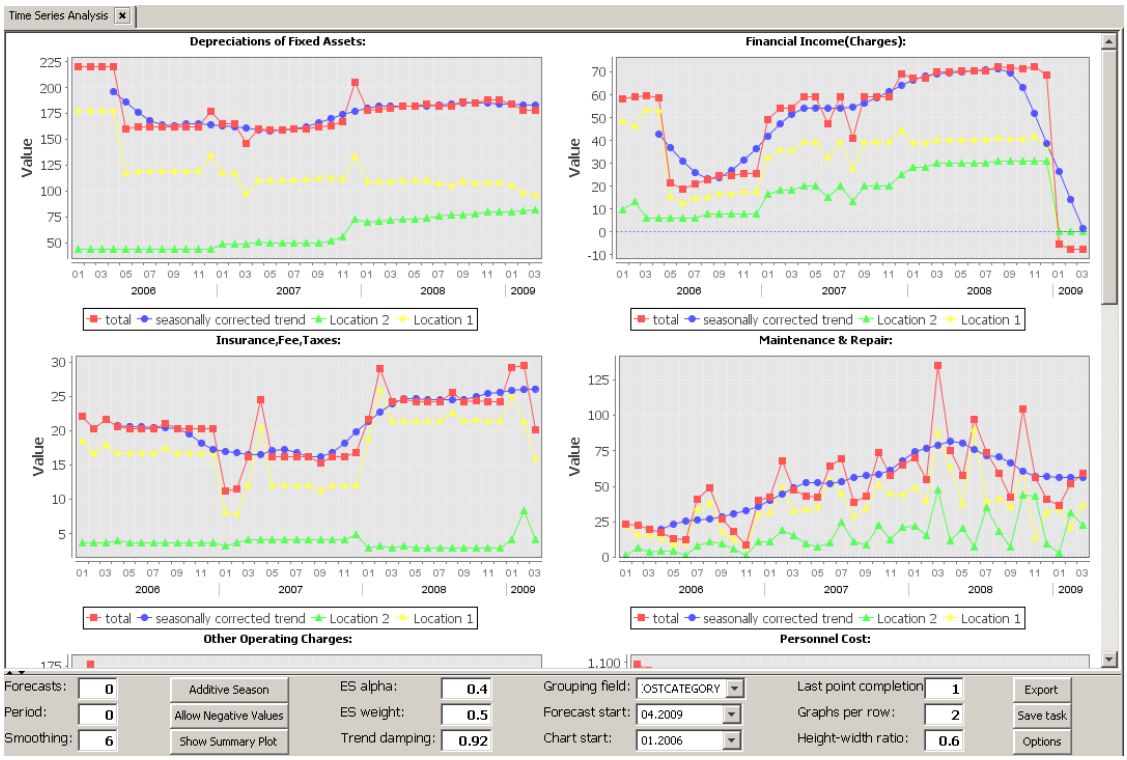
The **Time Series** tab has three vertically arranged regions:

- A **detail view** in which a separate time series chart for each value of the currently selected grouping field is shown (in our case `Location 1` and `Location 2`).
- A **global view** in which the total monthly cost is shown (red line), together with its seasonally corrected trend (blue line) and the percental distribution of total monthly cost among the two locations.
- A **tool bar** which permits to interactively work with the data, perform a trend analysis and calculate forecasts.

3.7.4 The detail plots

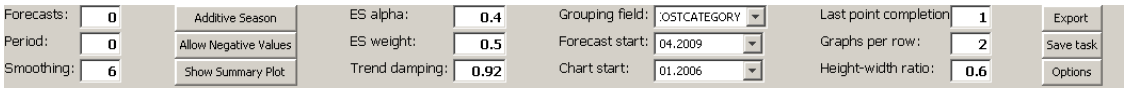
The upper part of the screen shows one single line chart for each value of the data field which has been selected as the **grouping field** in the toolbar. In example shown below, we have selected the field `CostCategory` as the grouping field; in this case, there is one single plot for each cost category.

Tip: mark a region inside one of the charts an drag a region with pressed left mouse button will enhance this region within all other charts.



3.7.5 The bottom tool bar

The displayed graphs and charts are depending on the settings in the tool bar.



Settings for calculating forecasts

- **Forecasts**:
  Number of forecasts, e.g. 3 for the following 3 periods (days, months, years etc.)
- **Period**:
  Presumed cycle length of the seasonal (periodic) part of the time series in units of the time step between adjacent data points. For example, if a yearly repeating pattern is presumed on monthly recorded data, enter 12 here.
- **Smoothing**:
  Number of time points for moving averages (trend lines). The trend lines are calculated as the symmetric moving average value of width 'Smoothing'. For example, if 'Smoothing' is 6, then the blue trend line values `tr(T)` at time point `T` are calculated from the red line values `v(T)` as `tr(T) = v(T-3)/2 + v(T-2) + v(T-1) + v(T) + v(T+1) + v(T+2) + v(T+3)/2`.

- **Additive / Multiplicative Season**:
  Multiplicative season means that the seasonal pattern is modeled as a correction factor to the long-term trend ('total = trend * season'). As a result, the amplitude of the seasonal fluctuation increases when the trend line increases, and it decreases when the trend line decreases. Additive season means that the seasonal pattern is modeled as an added term to the long-term trend ('total = trend + season'). As a result, the amplitude of the seasonal fluctuations is constant and does not grow when the trend line increases.

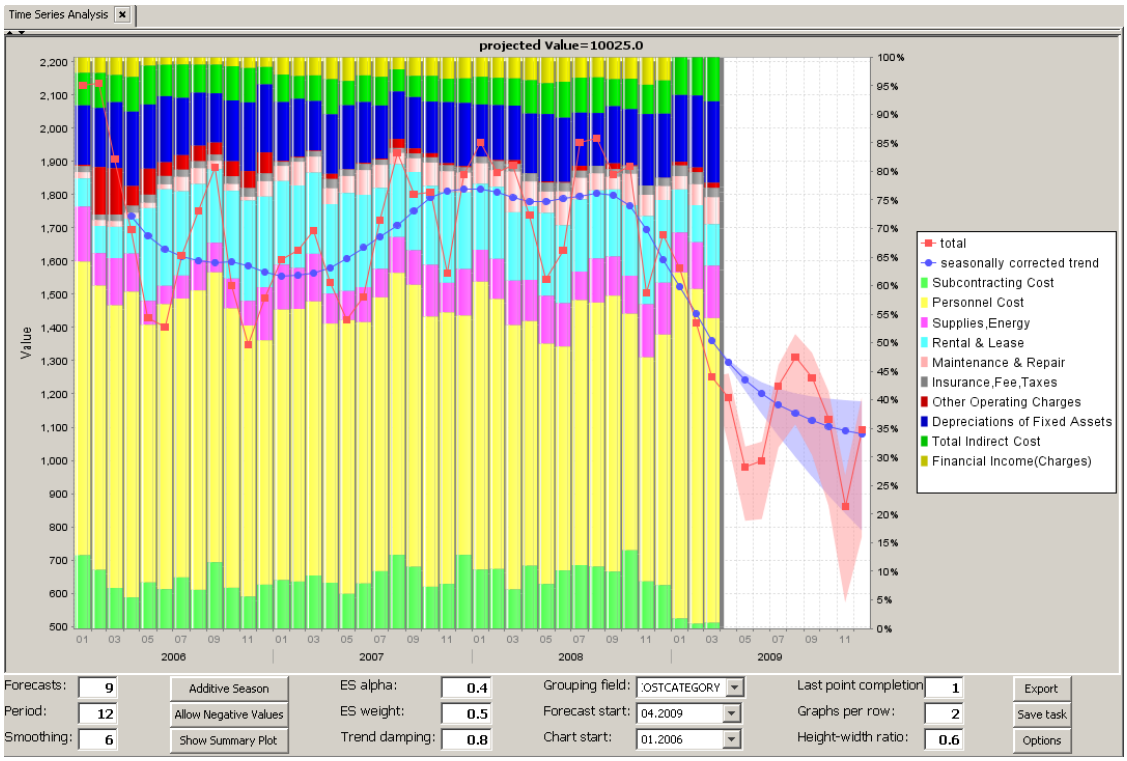
- **Allow Negative Values**:
  Specifies whether the predicted time series values can be negative or whether they will always be equal to or larger than zero.

- **ES alpha**:
  Exponential Smoothing coefficient alpha (defines a damping factor `(1-alpha)` per time step to the Exponential Smoothing contribution of the forecast.)

- **ES weight**:
  Weight prefactor to the Exponential Smoothing part of the forecast; `weight=0` switches off the Exponential Smoothing.

- **Trend damping**:
  Damping factor per time step. The damping factor is applied when projecting the current trend into the future.

In our example, data are available until March 2009 including. We want to create a forecast until end of the year, so 9 more montha. Furthermore, we see that the cost curve over the past years shows a cycle of 12 months. So we set the two parameters **Forecasts** and **Period** to the appropriate values and reduce the **trend damping** factor to 0.8:



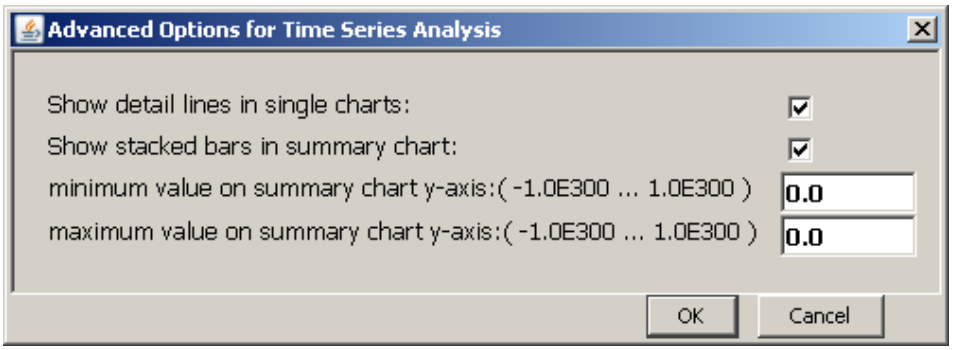
Basic settings for the chart generation

- **Show / Hide Summary Plot**:
  Activate/deactive the lower window part with the stacked bar chart.
- **Grouping field**:
  For each values of this field a separate detail chart is built.
- **Forecast start**:
  Starting time point for calculating the aggregated forecast values which are shown below the title line of each chart in the time series forecast screen.
- **Chart start**:
  First time point shown in the time series charts.
- **Last point completion**:
  Completion rate of the last time point, compared to the earlier time points. For example, if the last time point contains the aggregated sales amount of the first 14 out of 26 business days of the current months, the last point's completion should be set to 0.538 (= 14/26).
- **Graphs per row**:
  Number of detail graphs displayed per screen row.
- **Height-width ratio**:
  Height-Width-ratio of the detail charts.

Advanced settings for the chart generation

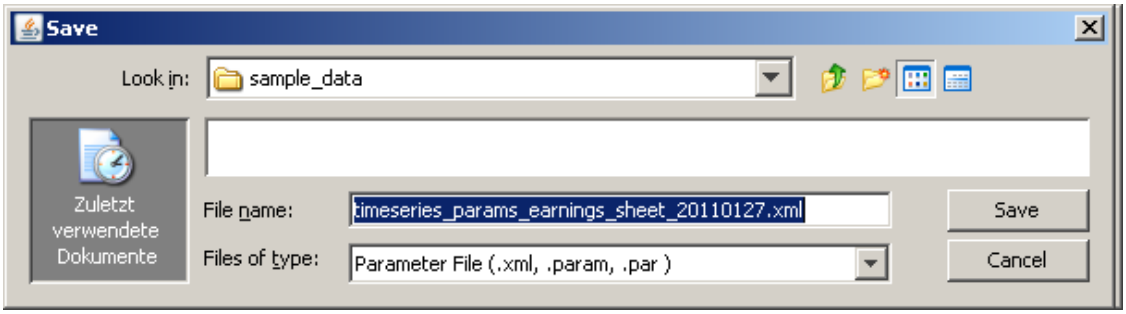
By clicking on the button **Options** in the toolbar you open a pop-up window which contains advanced options and settings for displaying the time series charts:

- **Show detail lines in single charts**:
  Activate/deactive the single value lines which appear below the total value line (red) and the total trend line (blue) in the detail charts for the various values of the grouping field.
- **Show stacked bars in summary chart**:
  Show the stacked bar diagram in the summary chart? Or only the summary lines (red and blue line plots)?
- **minimum / maximum value on summary chart y-axis**:
  Reduce the value range on the y-axis to a user-defined range.

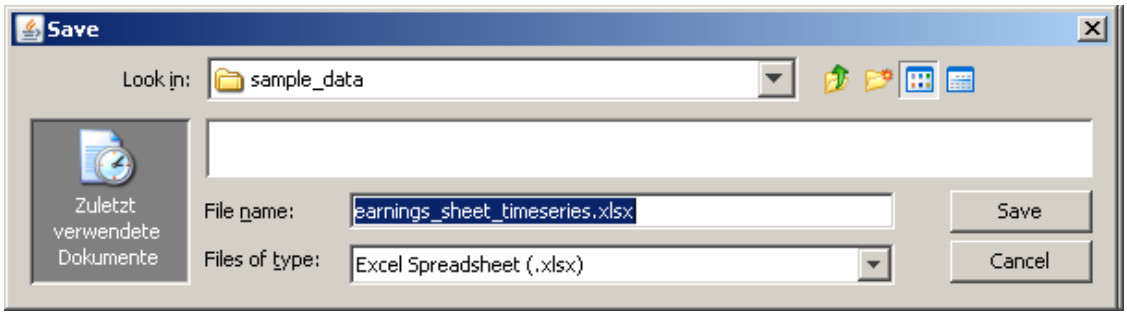

Advanced Options for Time Series Analysis
Show detail lines in single charts:
Show stacked bars in summary chart:
minimum value on summary chart y-axis:(-1.0E300 ... 1.0E300)
0.0
maximum value on summary chart y-axis:(-1.0E300 ... 1.0E300)
0.0
OK
Cancel

3.7.6 Saving and exporting settings and results

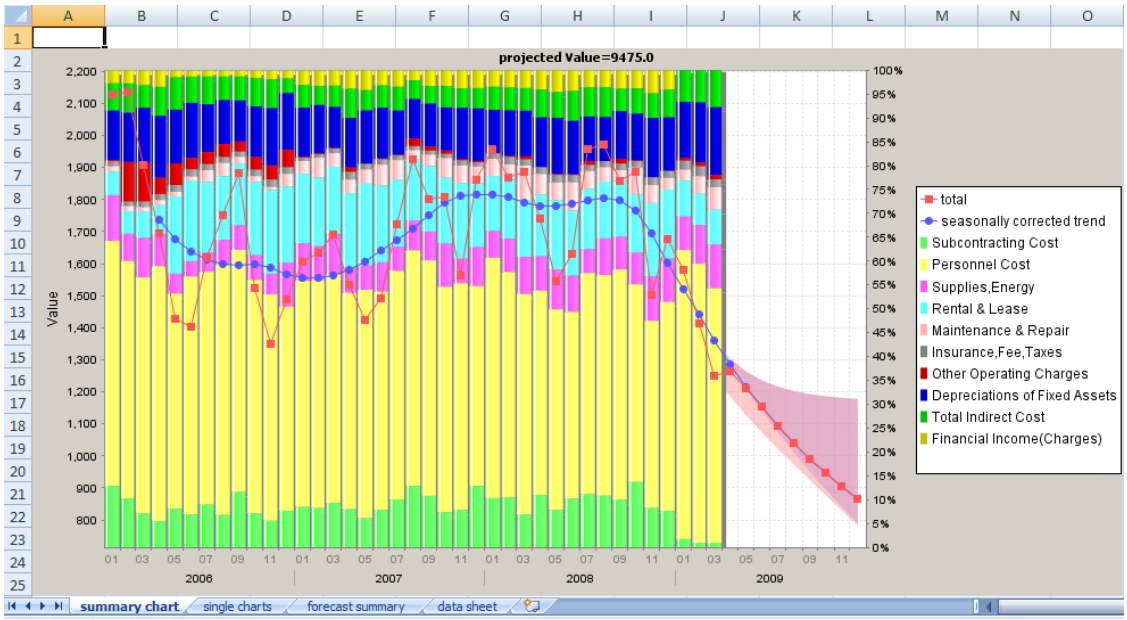
The analysis settings defined in the toolbar as well as the data import settings of the currently active data ource can be saved to a persistent parameter file by pressing the **Save task** button:



All data and charts may be exported to an Excel spreadsheet for further purposes:



The Excel file will contain separate tabs for each kind of result, i.e. summary chart, single charts, forecast summary, data sheet:



	A	B
1	Predicted accumulated values	
2	COSTCATEGORY	projected Value for 04.2009 - 12.2009
3	Subcontracting Cost	-865,0
4	Personnel Cost	6236,0
5	Supplies,Energy	1002,0
6	Rental & Lease	338,0
7	Maintenance & Repair	498,0
8	Insurance,Fee,Taxes	239,0
9	Other Operating Charges	90,0
10	Depreciations of Fixed Ass	1630,0
11	Total Indirect Cost	834,0
12	Financial Income(Charges)	-401,0
13	total Value	9475,4

	A	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS
1	Summ																						
2		2008											2009										
3		Subcontra	Personnel	Supplies,E	Rental & I	Maintena	Insurance	Other Ope	Depreciat	Total Indi	Financial	total	Subcontra	Personnel	Supplies,E	Rental & I	Maintena	Insurance	Other Ope	Depreciat	Total Indi	Financial	total
4	Location 1	1235,8	5826,4	987,6	1700,9	585,6	259,7	4,5	1298,2	756,7	480,5	13135,83	49,1	1359,6	214,2	63,3	90,7	62,2	23,4	299,2	182,3	-20,5	2323,5
5	Location 2	887,0	3778,0	571,8	1065,5	282,2	35,4	57,4	899,5	409,1	363,2	8349,19	1,8	1094,5	126,8	238,7	57,5	16,8	10,2	242,1	134,7	0,0	1923,1
6	total	2122,8	9604,4	1559,4	2766,4	867,8	295,1	61,9	2197,7	1165,8	843,7	21485,0	-814,4	8690,3	1342,8	639,5	646,1	317,6	123,5	2170,8	1151,3	-421,2	13722,0

3.8 Detecting Deviations and Inconsistencies

3.8.1 Purpose and short description

In the Deviation Detection panel, outliers, deviations and presumable data inconsistencies can be detected. The specific approach of this module is that it does not examine the values and value distribution characteristics of each data field separately for outliers as traditional data quality checker tools do. Rather, it finds cross-field inconsistencies.
For example, in a customer master data table neither the value `Age=35` nor the value `FamilyStatus=child` is an outlier or deviation, but the combination of both is one. This type of data errors are often overlooked by other data quality tools.

3.8.2 The result view

The deviations and inconsistency detection module was designed for usability for non-statisticians. It aims at delivering interesting results and findings without obliging the user to define hypotheses, busines rules or filter criteria and without too many 'expert' parameters and options. In the following, we are going to demonstrate a typical usage scenario of the module by means of an example analysis of the sample data `doc/sample_data/customers.txt`. We assume that these data have been read in as described in another chapter of this documentation, that means with `ClientID` as group field. If we start the module **Deviations and Inconsistencies** on these data and just press the **Start** button in the module's tool bar, we obtain the folloging result:

length	affected rec...	item supports	deviation stren...	item1	item2
2	1	{246;2135}	52.5	Age=..10[{=1(10)}	OnlineBanking=yes
2	6	{769;5135}	65.8	Age=[10..20[{=2(10)}	JointAccount=yes
2	1	{958;1320}	126.5	Age=[70..80[{=8(10)}	Profession=worker
2	2	{930;1396}	64.9	DurationClient=[25..29[{=8(10)}	Age=[20..30[{=3(10)}
2	1	{340;1732}	58.9	DurationClient=[29..33[{=9(10)}	Age=[30..40[{=4(10)}
2	1	{698;1904}	132.9	FamilyStatus=child	NumberDebits=[100..200[{=7(10)}
2	1	{698;1362}	95.1	FamilyStatus=child	AccountBalance=[10000..20000[{=8(10)}
2	4	{698;5135}	89.6	FamilyStatus=child	JointAccount=yes
2	2	{698;1919}	67.0	FamilyStatus=child	Age=[40..50[{=5(10)}
2	2	{698;1732}	60.4	FamilyStatus=child	Age=[30..40[{=4(10)}
2	2	{698;1541}	53.8	FamilyStatus=child	Profession=employee
2	1	{540;1584}	85.5	NumberCredits=[10..15[{=4(10)}	NumberDebits=[300..500[{=9(10)}
2	2	{1102;2027}	111.7	NumberCredits=[30..40[{=7(10)}	NumberDebits=..10[{=1(10)}
2	3	{1904;1959}	124.3	NumberDebits=[100..200[{=7(10)}	NumberCredits=..5[{=1(10)}
2	1	{440;1959}	86.2	NumberDebits=[50..70[{=5(10)}	NumberCredits=..5[{=1(10)}
2	1	{1727;1732}	299.1	Profession=retired	Age=[30..40[{=4(10)}
2	2	{1727;1919}	165.7	Profession=retired	Age=[40..50[{=5(10)}
2	1	{774;958}	74.1	Profession=technician,engineer	Age=[70..80[{=8(10)}

(Note: in the picture shown above, we have mouse-clicked the table column header of the column **item 1** in order to sort the detected deviations thematically, that means lexically). The different columns of the result table have the following meaning:

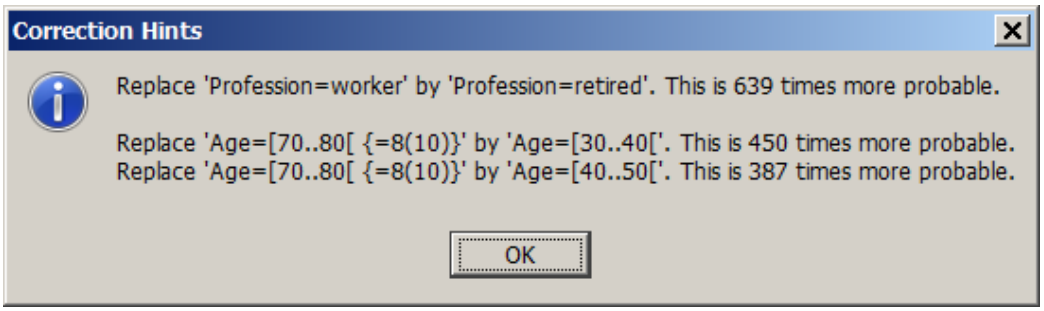
- **Length:**
  Length of the inconsistency pattern, i.e. the number of items in the pattern.

- **affected records:**
  Number of data records or groups on which the deviation pattern appears.
- **Item supports:**
  Number of data records or groups on which the different items which form the pattern appear.
- **Deviation strength:**
  The strength of a deviation pattern describes how strongly and significantly the number of occurrences of the pattern is below the expected number of occurrences.<p>The value is calculated as '10*(χ^2-conf - 0.9) / lift', where 'lift' is the pattern's lift and 'χ^2-conf' is the confidence level that the pattern is statistically significant.
  For example, if a combination (A,B) of two data field values A and B occurs in 0.02% of all records and has a χ^2 confidence level of 0.99, and if A and B alone occur in 20% respectively 10% of the data records, then the deviation strength of the pattern (A,B) is 90 since lift is 0.02%/(20%*10%) = 1/100 and 10*(χ^2-conf - 0.9) = 0.9.
- **Item 1, Item 2:**
  An item is an atomic part of an association or sequential pattern, i.e. a single piece of information, typically of the form [field name]=[field value] or [field name]=[field value range from ... to ...].

Hence, the deviation pattern which has been highlighted in blue in the above picture can be interpreted as follows: the combination of the two items `Age=[70 to 79 years]` (which is the 8th out of 10 value ranges of the date field `Age`) and `Profession=Worker` appears in one single data record. As the range `Age=[70 to 79 years]` appears in 958 out of 10000 data records and the value `Profession=Worker` in 1320 data records, we expected a much higher occurrence frequency, namely with about 958/10000 * 1320 = 126.5. That means, the lift value of the pattern is 1/126.5. The difference between the observed frequency of 1 and the expected frequency of 126.5 is highly significant (χ^2 confidence=1.000). The combination of the two, the confidence divided by the lift, is the deviation strength of the pattern (126.5).

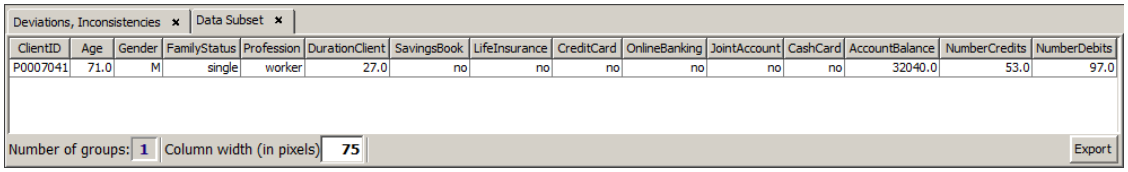
3.8.3 Obtaining correction hints

Some detected deviation patterns might be evident such as the pattern `Age=[30 to 49]` and`FamilyStatus=Child`. In many other detected patterns, however, it might be unclear which part of the pattern contains the part which does not fit with the rest of the pattern, and what replacement would most probably 'heal' the inconsistency or remove the data error. Synop Analyzer helps answering these questions by providing a 'correction hint' feature: richt-clicking on one of the patterns listed in the patterns table opens a pop-up window in which the software indicates which parts of the pattern are most probably the deviating parts and what are the statistically most probably corrections. The following picture shows the correction hint for the pattern which has been highlighted in the preceding screenshot:


Correction Hints
Replace 'Profession=worker' by 'Profession=retired'. This is 639 times more probable.
Replace 'Age=[70..80[{=8(10)}' by 'Age=[30..40['. This is 450 times more probable.
Replace 'Age=[70..80[{=8(10)}' by 'Age=[40..50['. This is 387 times more probable.
OK

The single correction hints displayed in the pop-up window are ordered by descending statistical plausibility. In the example shown above, the correction hint says that it would be normal if a person at an age of more than 70 years was not a worker but a pensioner. The second most plausible correction would be that the age of a person who has the profession 'worker' was between 30 and 50 years.

Often, it is advisable to check the displayed correction hints by looking at the involved data records. Then it often becomes obvious which one of the suggested corrections is the best matching one - or whether no corrections should be applied because the inflicted data sets are somehow untypical but not erroneous.Our example of the worker above 70 years occurs in one single data record:

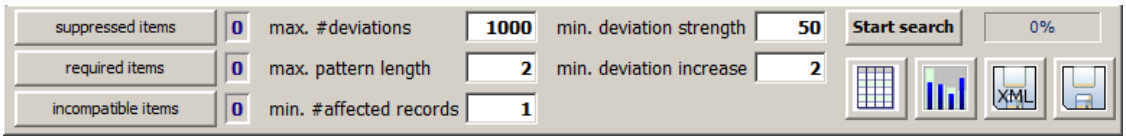

Deviations, Inconsistencies
Data Subset
ClientID Age Gender FamilyStatus Profession DurationClient SavingsBook LifeInsurance CreditCard OnlineBanking JointAccount CashCard AccountBalance NumberCredits NumberDebits
P0007041 71.0 M single worker 27.0 no no no no no no 32040.0 53.0 97.0
Number of groups: 1
Column width (in pixels) 75
Export

The closer inspection of this data set shows that most probably the value of the field `Profession` is outdated. The duration of the client relationship, the lack of adoption of 'modern' bank services (online banking, credit card, bank card) combined with an above-average account balance are more typical for a 71-year-old pensioner than for a younger worker.

3.8.4 The bottom tool bar

The tool bar at the lower edge of the panel provides features for

- modifying some analysis settings and thereby the obtained results,
- examining selected deviation patterns and the involved data records in more detail,
- permanently saving the analysis settings or the analysis results into an XML document, flat text file or spreadsheet.



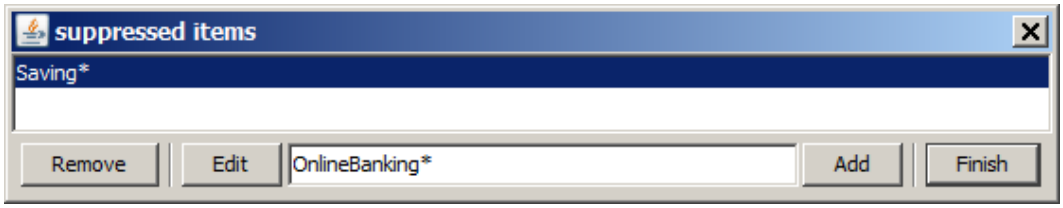
In the following we will describe these three groups of features in more detail.

Specification of the desired content of the deviation patterns

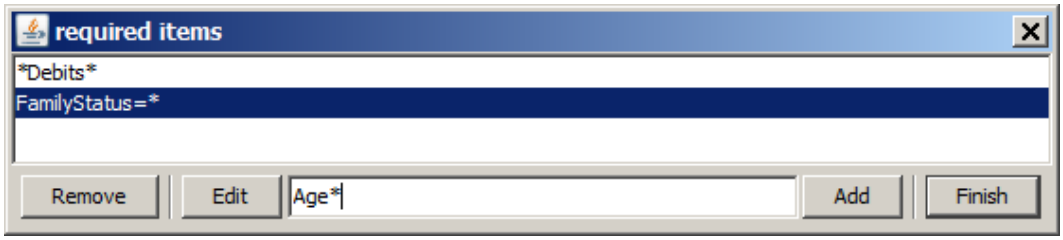
The three buttons at the left end of the tool bar help to focus the deviation detection to patterns with a user-specified content. Three kinds of specifications can be performed.

Using the button **Suppressed items** one can define groups of items which are to be completely ignored during the following deviation detection. Clicking on the button opens a pop-up window in which one can enter item names or parts of item names plus wildcard symbols (*) and activate each input by pressing the button **Add**. Each wildcard stands for zero or more arbitrary characters. You can either type in the desired values into the input field, or you can select from a drop-down list of all available items in the data by pressing the arrow symbol at the right edge of the input field.

In the screenshot below we have alredy specified that nothing involving the term `Saving` (as part of a field name or field value) should appear in the detected patterns. Then we have specified that we also want to suppress all patterns in which the term `OnlineBanking` occurs. This second limitation has not yet been activated by pressing the **Add** button.



Using the button **Required items** one can define groups of terms or items and enforce that each detected pattern contains at least one item from that group. Clicking on the button opens a pop-up window in which one can enter item names or parts of item names plus wildcard symbols (*) and activate each input by pressing the button **Add**. Each wildcard stands for zero or more arbitrary characters.



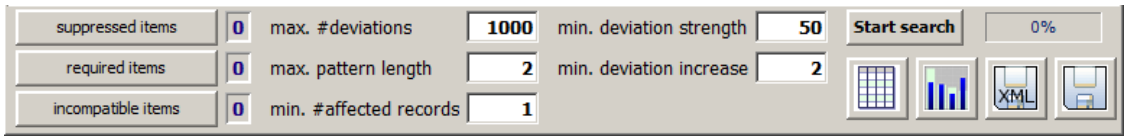
Using the button **Incompatible items** one can define pairs or tupels of items which must not occur together in the detected deviations. Clicking on the button opens a pop-up window in which one can enter combinations of item names or parts of item names plus wildcard symbols (*), separated by commas. Each entered combination must be activated by pressing the button **Add**. Each wildcard stands for zero or more arbitrary characters.

In the screenshot below we have specified that we do not want to find deviations which simulataneously contain values from the data fields `NumberCredits` and `NumberDebits`

The blue number fields at the right of the three aforementioned tool bar buttons indicate how many restrictions of the respective type have been defined and activated.

Modification of the statistical limits and settings

The five numeric input fields in the middle of the tool bar serve to specify desired value ranges for five statistical measures of the patterns to be detected:



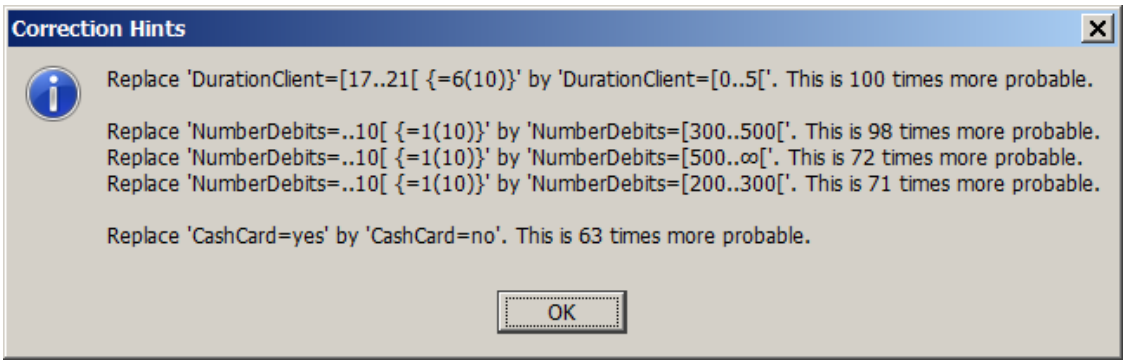
- **max #deviations:**
  Here you can specify an upper limit N for the number of deviation patterns to be found. If more patterns are detected which pass all other filters and specifications, only the N patterns with the highest deviation strengths will be displayed.
- **max. pattern length:**
  Here you can specify how many parts (items) the detected patterns may contain at maximum.
- **min. #affected records:**
  Here you can specify a lower limit for the number of data records (or data groups if a group field has been defined) on which the patterns to be detected must appear.
- **min. deviation strength:**
  Here you can the determine the required minimum deviation strength of all patterns to be detected.

- **min. deviation increase:**
  Here you can specify how strongly the deviation strength of a pattern of more than two parts (items) must exceed the deviation strengths of all its 'parent' patterns. A parent pattern is a pattern in which exactly one of the original pattern's item is missing. If you specify a value, make sure it is signifcantly larger than 1.0, for example 1.2, otherwise large numbers of pattern prologations of one single 'significant' short pattern can be displayed in which arbitrary items are appended to the significant short pattern.

The preceding picture shows an eample in which the predefined value of all five input fields has been modified. Additionally, the item content restrictions described in the preceding section have been maintained. Using these settings, Synop Analyzer finds the following deviation patterns:<A name="pattern3"/>

length	affected rec...	item supports	deviation stre...	item1	item2	item3
2	2	{1727;1919}	165.7	Profession=retired	Age=[40..50[{=5(10)}	
2	4	{698;5135}	89.6	FamilyStatus=child	JointAccount=yes	
3	2	{1411;2027;5065}	72.4	DurationClient=[17..21[{=...	NumberDebits=..10[{=1(1...	CashCard=yes
2	2	{698;1919}	67.0	FamilyStatus=child	Age=[40..50[{=5(10)}	
2	6	{769;5135}	65.8	Age=[10..20[{=2(10)}	JointAccount=yes	
2	2	{930;1396}	64.9	DurationClient=[25..29[{=...	Age=[20..30[{=3(10)}	
2	2	{698;1732}	60.4	FamilyStatus=child	Age=[30..40[{=4(10)}	
2	2	{698;1541}	53.8	FamilyStatus=child	Profession=employee	
2	9	{2027;2164}	48.7	NumberDebits=..10[{=1(1...	LifeInsurance=yes	
2	2	{698;1320}	46.1	FamilyStatus=child	AccountBalance=[20000..5...	
2	3	{246;5135}	42.1	Age=..10[{=1(10)}	JointAccount=yes	
2	3	{769;1584}	40.6	Age=[10..20[{=2(10)}	NumberDebits=[300..500[...	
3	2	{744;2027;5065}	38.2	FamilyStatus=widowed	NumberDebits=..10[{=1(1...	CashCard=yes
2	3	{769;1411}	36.2	Age=[10..20[{=2(10)}	DurationClient=[17..21[{=...	
2	3	{744;1396}	34.6	FamilyStatus=widowed	Age=[20..30[{=3(10)}	
2	2	{698;967}	33.7	FamilyStatus=child	CreditCard=yes	
3	3	{958;2027;5065}	32.8	Age=[70..80[{=8(10)}	NumberDebits=..10[{=1(1...	CashCard=yes
2	4	{744;1732}	32.2	FamilyStatus=widowed	Age=[30..40[{=4(10)}	
2	7	{967;2027}	28.0	CreditCard=yes	NumberDebits=..10[{=1(1...	
2	3	{813;958}	26.0	NumberDebits=[500.. {=1...	Age=[70..80[{=8(10)}	

Compared to the patterns created using the default settings, we notice that we now find some patterns of length 3. For these longer patterns, the 'correction hint' feature is often particularly helpful: the longer a pattern is, the more difficult is it to understand which part of the pattern does not match with the rest. In the screenshot below we show the correction hint for the pattern which has been highlighted in the picture above.

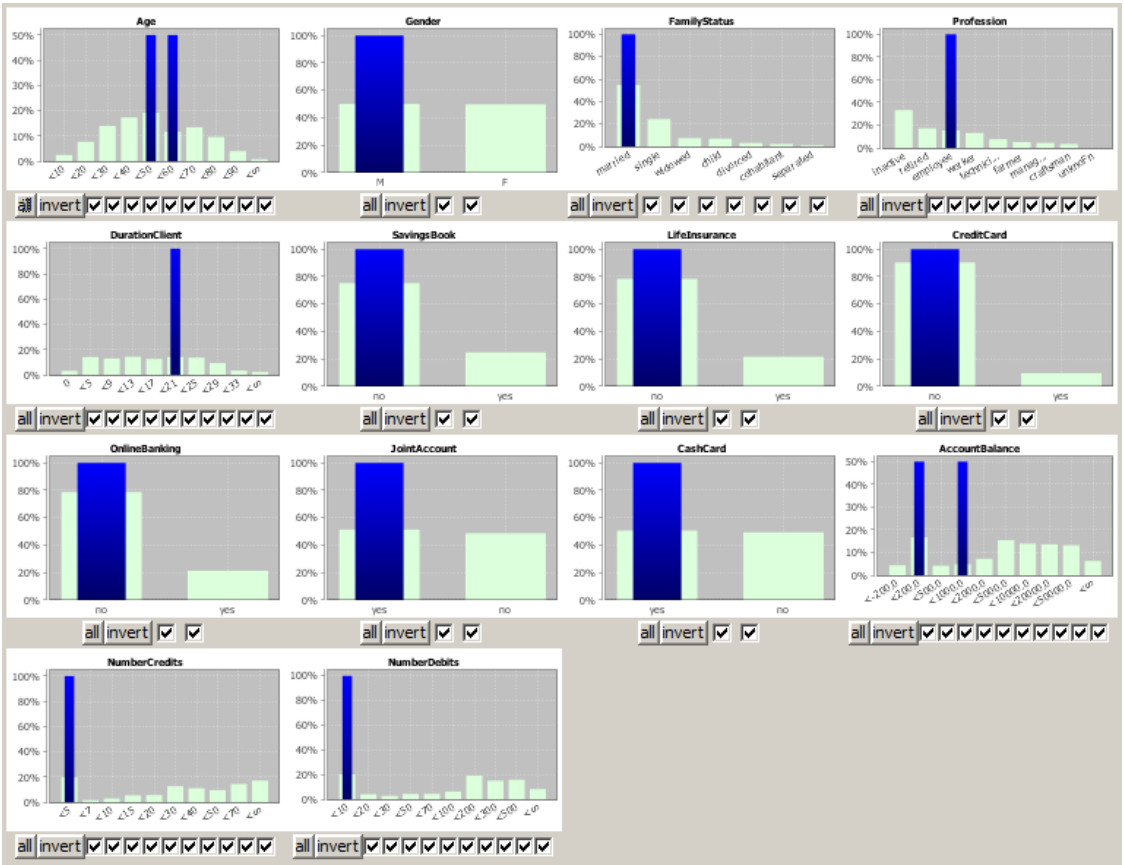

Correction Hints
Replace 'DurationClient=[17..21[{=6(10)}' by 'DurationClient=[0..5['. This is 100 times more probable.
Replace 'NumberDebits=..10[{=1(10)}' by 'NumberDebits=[300..500['. This is 98 times more probable.
Replace 'NumberDebits=..10[{=1(10)}' by 'NumberDebits=[500..∞['. This is 72 times more probable.
Replace 'NumberDebits=..10[{=1(10)}' by 'NumberDebits=[200..300['. This is 71 times more probable.
Replace 'CashCard=yes' by 'CashCard=no'. This is 63 times more probable.
OK

From the correction hints we understand what makes this pattern a deviation: long-term inactive (nominal) clients should not have an active bank card. The item `BankCard=yes` is the one which does not fit into the rest of the pattern.

Detailed introspection of selected deviation patterns

Another possibility of deviation pattern introspection is to compare the properties of the data records which are affected by the selected pattern(s) to the entire data. This can be done using the multivariate exploration technique known from the module Multivariate Exploration, but with the affected data records as fixed preselection. By deactivating some of the checkboxes below the histogram charts you can further reduce the data selection.

This function is provided via the button **Explore selected** on the right side of the toolbar. If you press that button after selecting the pattern of length 3 which has been discussed in the previous section, you obtain the following result:



We notice that the affected customers are mainly married male employees between 40 and 60 years which are long-term customers and have a joint account and a bank card. This is a very normal, unremarkable combination. More noticeable is the fact that their accounting activity (`NumberDebits` and `NumberCredits` is close to zero, which is quite untypical for this customer group.

Our preliminary result is that the examined pattern does probably not indicate data errors: apart from the accounting activity, all demographic data properties of the affected records are consistent. The question now is which of the involved customers are purely nominal clients which should be removed from the customer master data because they generate negative margins, and which customers could and should be 'reactivated'.

In order to answer this question, we use another tool bar function: the button **Show**. This button opens a new window in which those data records are shown which are affected by at least one of the currently selected deviation patterns.

Note: A table of the affected data records can also be opened from the multivariate exploration pop-up window by clicking on the button **Show** in the tool bar of that window. This second option has the advantage that one can hide and suppress a part of the data fields by means of the button **Visible fields** before opening the tabular data view. In contrast, pressing the button **Show** on the main toolbar of the deviation detection panel always shows all data fields of the displayed data records.

If only the one pattern of length 3 has been selected which has been discussed above, the tabular data records view looks as follows:

ClientID	Age	Gender	FamilyStatus	Profession	Duration...	SavingsBook	LifeIns...	CreditCard	OnlineB...	JointAc...	CashCard	AccountBal...	NumberCr...	NumberD...
P0031522	52.0	M	married	employee	19.0	no	no	no	no	yes	yes	966.8	2.0	5.0
P0034770	41.0	M	married	employee	18.0	no	no	no	no	yes	yes	-8.63	0.0	1.0

From this introspection we understand that the second customer, `P0034770` probably belongs to the category 'nominal client': during one year, the customer had no credit transaction and only one debit transaction, probably an account-keeping fee so that the account balance has slipped into the slightly negative range. This customer generates most probably more cost than profit, and a 'reactivation' is highly improbable.

The first customer, `P0031522`, on the contrary, shows some financial activity on his accounts. Here, trying to reactivate the customer might be more promising.

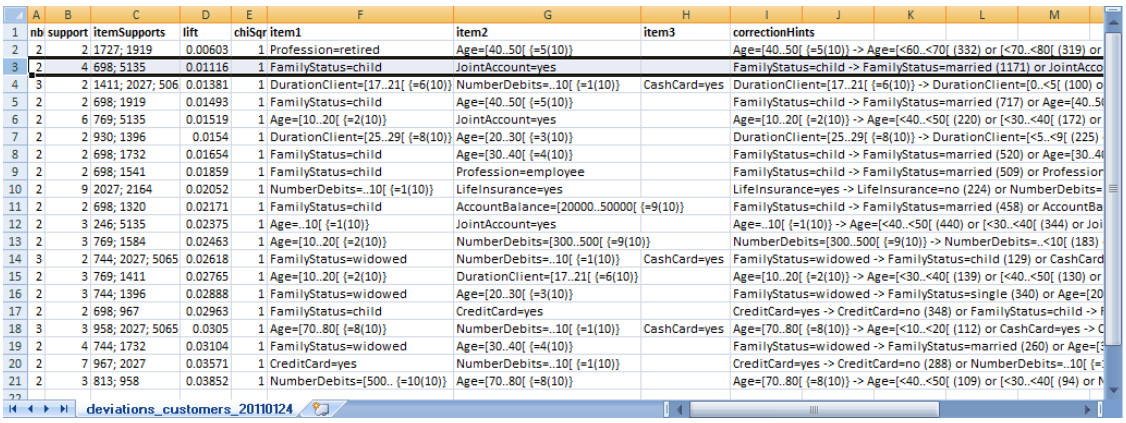
Saving and exporting results

At the end of a data analysis one often wants to permanently save the analysis settings, or to export the analysis results so that they can be used outside of Synop Analyzer. The tool bar of the module **Deviations and Inconsistencies** offers four functions for achieving this:

1.

:
By means of this button one can save the currently active settings for this module and for importing the data to an XML parameter file. The structure of this file conforms to the XML schema http://http://www.synop-systems.com/xml/InteractiveAnalyzerTask.xsd. This file can later be reloaded via the main menu button **Analysis → run Deviations and Inconsistencies**. This starts a process which reads the most current version of the data specified in the XML file and then starts the module **Deviations and Inconsistencies** with the settings stored in the XML file.

2. :
This button saves the deviation patterns which are currently displayed in the main part of the panel into a `TAB` separated flat text file which can be opened in any text editoror with MS Excel. The results shown in the section Modifying the statistical settings look like this when exported and re-opened in MS Excel:



nb	support	itemSupports	lift	chiSqr	item1	item2	item3	correctionHints
2	2	1727; 1919	0.00603	1	Profession=retired	Age=[40..50[{=5(10)}		Age=[40..50[{=5(10)} -> Age=[<60..<70[(332) or [<70..<80[(319) or
2	4	698; 5135	0.01116	1	FamilyStatus=child	JointAccount=yes		FamilyStatus=child -> FamilyStatus=married (1171) or JointAcco
3	2	1411; 2027; 506	0.01381	1	DurationClient=[17..21[{=6(10)}	NumberDebits=..10[{=1(10)}	CashCard=yes	DurationClient=[17..21[{=6(10)} -> DurationClient=[0..<5[(100) o
2	2	698; 1919	0.01493	1	FamilyStatus=child	Age=[40..50[{=5(10)}		FamilyStatus=child -> FamilyStatus=married (717) or Age=[40..5
2	6	769; 5135	0.01519	1	Age=[10..20[{=2(10)}	JointAccount=yes		Age=[10..20[{=2(10)} -> Age=[<40..<50[(220) or [<30..<40[(172) or
2	2	930; 1396	0.0154	1	DurationClient=[25..29[{=8(10)}	Age=[20..30[{=3(10)}		DurationClient=[25..29[{=8(10)} -> DurationClient=[<5..<9[(225)
2	2	698; 1732	0.01654	1	FamilyStatus=child	Age=[30..40[{=4(10)}		FamilyStatus=child -> FamilyStatus=married (520) or Age=[30..4
2	2	698; 1541	0.01859	1	FamilyStatus=child	Profession=employee		FamilyStatus=child -> FamilyStatus=married (509) or Professior
2	9	2027; 2164	0.02052	1	NumberDebits=..10[{=1(10)}	LifeInsurance=yes		LifeInsurance=yes -> LifeInsurance=no (224) or NumberDebits=
2	2	698; 1320	0.02171	1	FamilyStatus=child	AccountBalance=[20000..50000[{=9(10)}		FamilyStatus=child -> FamilyStatus=married (458) or AccountBa
2	3	246; 5135	0.02375	1	Age=..10[{=1(10)}	JointAccount=yes		Age=..10[{=1(10)} -> Age=[<40..<50[(440) or [<30..<40[(344) or Joi
2	3	769; 1584	0.02463	1	Age=[10..20[{=2(10)}	NumberDebits=[300..500[{=9(10)}		NumberDebits=[300..500[{=9(10)} -> NumberDebits=..<10[(183)
3	2	744; 2027; 5065	0.02618	1	FamilyStatus=widowed	NumberDebits=..10[{=1(10)}	CashCard=yes	FamilyStatus=widowed -> FamilyStatus=child (129) or CashCard
2	3	769; 1411	0.02765	1	Age=[10..20[{=2(10)}	DurationClient=[17..21[{=6(10)}		Age=[10..20[{=2(10)} -> Age=[<30..<40[(139) or [<40..<50[(130) or
2	3	744; 1396	0.02888	1	FamilyStatus=widowed	Age=[20..30[{=3(10)}		FamilyStatus=widowed -> FamilyStatus=single (340) or Age=[20
2	2	698; 967	0.02963	1	FamilyStatus=child	CreditCard=yes		CreditCard=yes -> CreditCard=no (348) or FamilyStatus=child -> F
3	3	958; 2027; 5065	0.0305	1	Age=[70..80[{=8(10)}	NumberDebits=..10[{=1(10)}	CashCard=yes	Age=[70..80[{=8(10)} -> Age=[<10..<20[(112) or CashCard=yes -> C
2	4	744; 1732	0.03104	1	FamilyStatus=widowed	Age=[30..40[{=4(10)}		FamilyStatus=widowed -> FamilyStatus=married (260) or Age=[3
2	7	967; 2027	0.03571	1	CreditCard=yes	NumberDebits=..10[{=1(10)}		CreditCard=yes -> CreditCard=no (288) or NumberDebits=..10[{=
2	3	813; 958	0.03852	1	NumberDebits=[500.. {=10(10)}	Age=[70..80[{=8(10)}		Age=[70..80[{=8(10)} -> Age=[<40..<50[(109) or [<30..<40[(94) or N

deviations_customers_20110124

The exported version of the patterns differs in three points from the version shown on screen. First, non-localized english column names are used. Second, instead of the column 'deviation strength' the two values are exported from which the deviation strength is calculated: the patterns' `lift` and `chiSqrConfidence`. Third, an additional column is added which contains a slightly shortened version of the correction hints which appear in a separate pop-up window in the on-screen version of the patterns.

3. :
The pop-up window 'show data records' has its own export button with which all data records on which at least one of the selected deviation patterns appears are exported into a `TAB` separated flat text file.

4. → **Export :**
Similarly, the pop-up window 'multivariate exploration' contains its own export button. This button exports all data records on which at least one of the selected deviation patterns appears, plus all multivariate histogram chart plots into a spreadsheet file in the `.xlsx` format.n

3.8.5 Interpretation of deviations: untypical data set or data error?

The module 'Deviations and Inconsistencies' shows data records and patterns which are significantly untypical. If the module is used for data quality monitoring purposes, two questions have to be answered for each detected pattern and each affected data record:

1. Does the affected data record contain an data fault which should be removed, or are the data correct and they just describe something untypical?

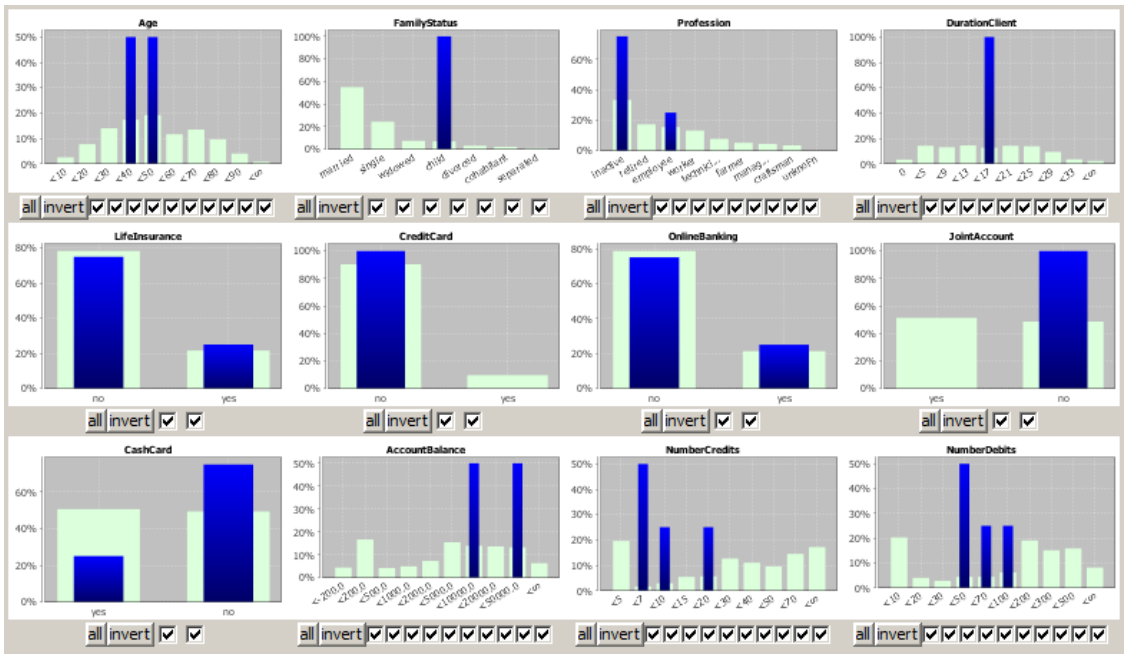
2. If the data record contains a data fault, which data field contains the faulty value, and what is the correct value?n

The module contains a couple of tools for answering these questions - the correction hints, the multivariate exploration and the data record introspection which have been described in earlier sections of this chapter. However, it should be noted that the correction hint can be misleading in some situations, and an automatic data correction process based on these correction hints and without further human controll is not advisable.

After this initial remark we want to revisit two of the examples discussed above. In these examples, a human introspector quickly understands that they must be data faults:

length	affected records	item supports	deviation strength	item1	item2
2	1	{246;2135}	52.5	Age=..10[{=1(10)}	OnlineBanking=yes
2	6	{769;5135}	65.8	Age=[10..20[{=2(10)}	JointAccount=yes
2	1	{958;1320}	126.5	Age=[70..80[{=8(10)}	Profession=worker
2	2	{930;1396}	64.9	DurationClient=[25..29[{=8(10)}	Age=[20..30[{=3(10)}
2	1	{340;1732}	58.9	DurationClient=[29..33[{=9(10)}	Age=[30..40[{=4(10)}
2	1	{698;1904}	132.9	FamilyStatus=child	NumberDebits=[100..200[{=7(...
2	1	{698;1362}	95.1	FamilyStatus=child	AccountBalance=[10000..20000...
2	4	{698;5135}	89.6	FamilyStatus=child	JointAccount=yes
2	2	{698;1919}	67.0	FamilyStatus=child	Age=[40..50[{=5(10)}
2	2	{698;1732}	60.4	FamilyStatus=child	Age=[30..40[{=4(10)}
2	2	{698;1541}	53.8	FamilyStatus=child	Profession=employee
2	1	{540;1584}	85.5	NumberCredits=[10..15[{=4(10)}	NumberDebits=[300..500[{=9(...
2	2	{1102;2027}	111.7	NumberCredits=[30..40[{=7(10)}	NumberDebits=..10[{=1(10)}
2	3	{1904;1959}	124.3	NumberDebits=[100..200[{=7(...	NumberCredits=..5[{=1(10)}
2	1	{440;1959}	86.2	NumberDebits=[50..70[{=5(10)}	NumberCredits=..5[{=1(10)}
2	1	{1727;1732}	299.1	Profession=retired	Age=[30..40[{=4(10)}
2	2	{1727;1919}	165.7	Profession=retired	Age=[40..50[{=5(10)}
2	1	{774;958}	74.1	Profession=technician,engineer	Age=[70..80[{=8(10)}

Since children of more than 21 years do not exist in any country on earth, we are faced with the question which values are faulty in the affected data sets. In order to answer this question we first look at the multivariate exploration of the four data records in which children of more than 21 years appear. From this graphical data exploration one often gets hints on where the data fault is located, for example could all affected data records carry one identical data import time stamp or they could stem from one identical source system or one filiation etc.



In our example, we get the impression that the four customers show rather a typical adult behaviour than a typical child behaviour (see the fields `DurationClient`, `AccountBalance` or `NumberCredits`. Now we introspect the data records themselves:

ClientID	Age	FamilyS...	Profession	Duration...	LifeInsur...	CreditCard	OnlineBa...	JointAcc...	CashCard	Account...	NumberCredits	NumberDebits
P0041881	48.0	child	inactive	16.0	no	no	no	no	no	25890.0	6.0	43.0
P0044710	34.0	child	employee	15.0	no	no	no	no	no	6498.0	9.0	56.0
P0044861	33.0	child	inactive	15.0	yes	no	yes	no	yes	8623.0	19.0	78.0
P0098570	47.0	child	inactive	15.0	no	no	no	no	no	27180.0	6.0	43.0

The first impression is consolidated: in each single data record we find three to four indications for the person being an adult (see `Profession`, `AccountBalance`, `BankCard`, `NumberCredits`). As a human processor we could now delete the four values `FamilyStatus=child` and either replace them by `unknown` or send the data records to a colleague who gathers the correct family status data.

3.9 The Associations Analysis module

3.9.1 Purpose and short description

An associations analysis finds items in your data that are associated with each other in a meaningful way. Within Synop Analyzer, an associations analysis is started by pressing the button in the left screen column.

An item is an 'atomic' piece of information contained in the input data, that means a combination of a data field name and a field value, for example `PROFESSION=farmer`. A prerequisite for finding associations between these atomic items is that a grouping of several of the items to one comprising group of data fields or data records exists. Often, this group of fields or records is called a transaction (TA).

An association is a combination of several items, a so-called item set, for example the combination `PROFESSION=farmer & GENDER=male`. An association rule is a combination of two item sets in the form of a rule `itemset1` → `itemset2`. The left hand side of the rule is called the rule body or antecedent, the right hand side the rule head or consequent.

The table below lists typical use cases for associations analysis [Ballard, Rollins, Dorneich et al., Dynamic Warehousing: Data Mining made easy]:

industry	**use case**	**grouping criterion**	**typical body item**	**typical head item**
retail	market basket analysis	bill ID or purchase ID	a purchased article	another purchased article
manufacturing	quality assurance	product (e.g. vehicle ID)	component, production condition	problem, error ID
medicine	medical study evaluation	patient or test person	single treatment info	medical impact

3.9.2 Input data formats

Synop Analyzer's association detection module is prepared for working with three different data formats:

- **The transactional or pivoted data format:**
  Often, the input data for associations mining are available in a format in which one column is the so-called group field and contains transaction IDs, one or more additional fields are the so-called item fields and contain items, i.e. the information on which associations are to be detected.
  Synop Analyzer expects that data with a group field are sorted by group field values. If the data are read from a database, Synop Analyzer automatically assures that property by issuing a SELECT statement with an appropriate ORDER BY clause.

If the data are read from flat file or from a spreadsheet, the user is responsible for bringing the data into the correct order. Synop Analyzer will issue a warning message if the data are not correctly ordered.
The file doc/sample_data/RETAIL_PURCHASES.txt is an example for such a data format: the field PURCHASE_ID is the group field, the field ARTICLE contains the real information, namely the IDs of the purchased articles.
In the transactional data format, the items appearing in the detected association patterns are a combination of field name and field value if there is more than one item field; the name of the item field is omitted if all items come from one single field (such as the field ARTICLE).

- **The data format with Boolean fields:**
  You can also detect associations on input data which do not have a group field (that means each data row represents a separate transaction) and in which each single 'item', i.e. each single event or fact, has its own two-valued (Boolean) data field which indicates whether or not the item occurs in the transaction.
  If the field PURCHASE_ID was missing in the sample data doc/sample_data/RETAIL_PURCHASES.txt and if there was a separate data field for each existing article ID which contained either 0 or 1, depending on whether or not the corresponding article was purchased in transaction represented by the current data row, then the data would have the data format with Boolean fields.
  If Synop Analyzer detects a data format with Boolean fields, it interprets all Boolean field values starting with '0', '-', 'F' or 'f' (such as 'false'), 'N' or 'n' (such as 'no' or 'n/a') as indicators for 'item does not occur in the transaction', all other values are interpreted as 'item occurs in the transaction'.
  In the data format with Boolean fields the items appearing in the detected association patterns contain only the names of the Boolean fields, but not the field values such as 'YES' or '1'.

- **The 'normal' or broad data format:**
  Of course, Synop Analyzer can also detect associations on 'normal' data in which each single data row is considered one data group and in which there are different data fields of various types which contain the items. doc/sample_data/customers.txt is an example for such a file.
  On these data, the items appearing in the detected patterns always have the form 'field_name=field_ value'.

A general rule, which is valid on all data formats, is: the items which form the detected associations can only come from active data fields which have not been marked as 'group', 'entity', 'oder' or 'weight'. 'entity' fields are ignored in associations mining (they are only important for sequential patterns analysis), 'group' field values serve to define data groups covering more than one data row, information from 'order' fields is used to calculate trend coefficients for the detected associations, and information from 'weight' fields is used to calculate pattern weight coefficients.

3.9.3 Definitions and notations

An association pattern or rule can be characterized by the following properties: [Ballard, Rollins, Dorneich et al., Dynamic Warehousing: Data Mining made easy]

- The **items**
  which are contained in the rule body, in the rule head, or in the entire rule.
- **Categories** of the contained items.
  Often, an additional hierarchy or taxonomy for the items is known. For example, the items 'milk' and 'baby food' might belong to the category 'food', 'diapers' might belong to the category 'non-food'. 'axles=17-B' and 'engine=AX-Turbo 2.3' might be members of the category 'component', 'production_chain=3' of category 'production condition', and 'delay > 2 hours' of category 'error state'. Hence, the second sample rule can be characterized by the fact that its body contains components or production conditions, and its head an error state.
- The **support** of the pattern:
  Absolute support `S` is defined as the total number of data groups (transactions) for which the rule holds. Support or relative support `s` is the fraction of all transactions for which the rule holds.
- The **confidence** of the pattern when interpreted as a rule:
  Confidence `C` is defined as
  `C := s(body & head) / s(body).`
- The **lift** of the pattern:
  The Lift `L` of a pattern $\texttt{Item}_1$ `&` ... `&` $\texttt{Item}_n$ is defined as
  $\texttt{L := s(Item}_1\texttt{ \& ... \& Item}_n\texttt{) / (s(Item}_1\texttt{) * ... * s(Item}_n\texttt{))}$.
  lift > 1 (< 1) means that the pattern appears more (less) frequently than expected assuming that all involved items are statistically independent.
- The **rule lift** of the pattern when interpreted as a rule:
  The rule lift $\texttt{L}_r$ is defined as
  $\texttt{L}_r\texttt{ := s(body \& head) / (s(body) * s(head))}$.
  $\texttt{L}_r > 1$ (< 1) means that the pattern `body & head` appears more (less) frequently than expected assuming that `body` and `head` are statistically independent.
- The **purity** of the pattern:
  The purity `P` of a pattern $\texttt{Item}_1$ `&` ... `&` $\texttt{Item}_n$ is defined as
  $\texttt{P := s(Item}_1\texttt{ \& ... \& Item}_n\texttt{) / max}_{i=1\ldots n}\texttt{(s(Item}_i\texttt{))}$.
  $\texttt{P} = 1$ means that the pattern describes a 'perfect tupel': none of the items $\texttt{Item}_i$ ever occurs without all the other items in the same data group.
- The **core purity** of the pattern:
  The core purity $\texttt{P}_c$ of a pattern $\texttt{Item}_1$ `&` ... `&` $\texttt{Item}_n$ is defined as
  $\texttt{P}_c\texttt{ := s(Item}_1\texttt{ \& ... \& Item}_n\texttt{) / min}_{i=1\ldots n}\texttt{(s(Item}_i\texttt{))}$.
  $\texttt{P}_c = 1$ means that at least one of the items involved in the pattern does never occur in the data without the `n-1` other items of the pattern. An item with this property is called a 'core item' of the pattern.

- The **weight (cost,price)** of the pattern:
  If a weight field has been defined on the input data, we can calculate the average weight of the data groups which support the pattern. If, for example, in those purchases which contain the items 'milk', 'baby food' and 'diapers', the mean overall purchase value is 49.69, *theweightofthepattern*(`milk & baby food & diapers`), *andalsotheweig*

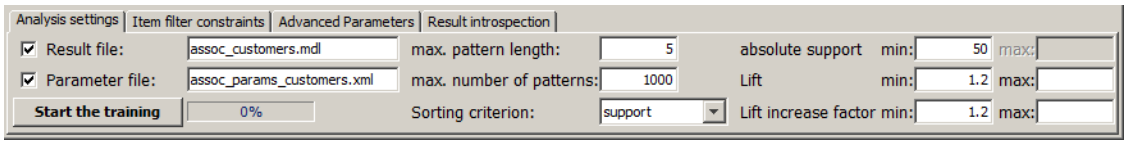
- The χ^2 **confidence** of the pattern:
  The χ^2 confidence level of an association indicates up to which extent each single item is relevant for the association because its occurrence probability together with the other items of the association significantly differs from its overall occurrence probability.
  More formally, the χ^2 confidence level is the result of performing n χ^2 tests, one for each item of the association. The null hypothesis for each test is: the occurrence frequency of the item is independent of the occurrence of the item set formed by the other n-1 items.
  Each of the n tests returns a confidence level (probability) with which the null hypothesis is rejected, and the χ^2 confidence level of the association is set to the minimum of these n rejection confidences.

3.9.4 Basic parameters for an Associations analysis

In Synop Analyzer, an associations analysis is started by loading a data source - the so-called training data - into memory and by clicking on the button in the input data panel on the left side of the Synop Analyzer GUI. The button opens a panel named **Associations Detection**. In the lower part of this panel, you can specify the settings for an associations analysis and start the search. The detection process itself can be a long-running task, therefore it is executed asynchronically in several parallelized background threads. In the upper part of the panel, the detected association rules - the so-called association model - are displayed.

The following paragraphs and screenshots demonstrate the handling of the various sub-panels and buttons at hand of the sample data `doc/sample_data/customers.txt`.

The first visible tab in the toolbar at the lower end of the screen contains the most important parameters for associations analysis.



In the screenshot, the following settings were specified:

- The detected association patterns will be saved under the name `assoc_customers.mdl` in the current working directory. Per default, the created file will be a file in a proprietary binary format. But you could also save the file as a <TAB> separated flat text file, which can be opened in any text editor or spreadsheet processor such as MS Excel. Using the main menu item **Preferences→Associations Preferences** you can switch the output format, for example to the intervendor XML standard for data mining models, PMML.

- The currently specified settings will automatically be saved to an XML parameter file named `assoc_params_customers.xml` every time the button **Start training** will be pressed. The resulting XML file can be reloaded in a later Synop Analyzer session via the main menu item **Analysis→Run associations analysis**. This reproduces exactly the currently active parameter settings and data import settings.

- The patterns to be detected should consist of up to 5 parts (items). When specifying the parameters for an associations training, you should always specify an upper boundary for the desired association lengths, otherwise the training can take extremely long time.

- The upper limit for the number of patterns to be detected and displayed is set to 1000. If more patterns are found, the 1000 patterns with the highest values of the measure currently specified in the selector box **Sorting criterion** will be selected. In our example, the 1000 patterns with highest support will be selected.

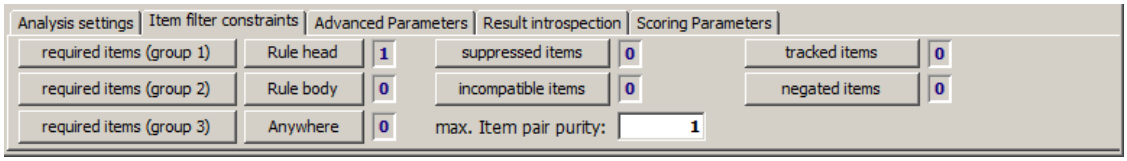
- The patterns to be detected should occur in at least 50 data groups (transactions). When specifying the parameters for an associations training, you should always specify an lower boundary for the absolute or relative support, otherwise the training can take extremely long time.

- Only patterns whose lift is at least 1.2 are to be detected. Hence, we are interested only in 'frequent' patterns which appear on at least 20% more data groups than it could have been expected from the frequencies of the involved items.

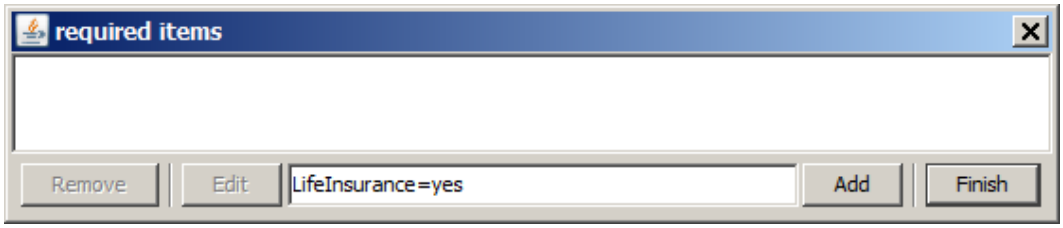
- The patterns consisting of more than two parts (items) must have lift increase factors of at least 1.2. An association pattern of n $>$ 2 items has n lift increase factors, namely the patterns own lift value divided by the n lift values of the n 'parent' patterns in which exactly one of the n items is missing.
  The specification of an upper or lower limit for the lift increase factor often is a very effective means for preventing the set of detected patterns from growing too big and for suppressing the appearance of 'redundant', trivial extensions of relevant patterns by just appending arbitrary items to them. As a general rule, one should always specify a minimum value larger than 1 for both lift and lift increase factor if one is looking for typical, frequent patterns. On the other hand, if one is looking for deviations, one should always specify a maximum lift and maximum lift increase factor smaller than 1.

3.9.5 Pattern content constraints ('item filters')

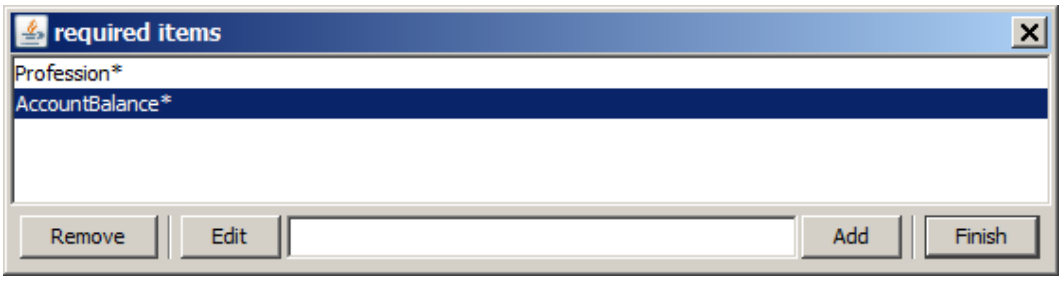
Filter criteria defining the desired contant of the patterns to be detected can be specified using the second tab named **Item filters** of the bottom part of the associations analysis screen. The tab itself displays how many content filter criteria of the various types have been set, the specification of new content filter criteria is performed within pop-up dialogs which open up when one presses one of the buttons in the tab.



- The three buttons named **Required items (group n)** define items which must occur in each detected pattern. If several item patterns are specified within one 'required group', at least one of them must appear in each detected association. In the Associations analysis module, up to 3 different groups of required items can be specified. The detected patterns must contain at least one item out of every specified group.
  Each item specification can contain wildcards (*) at the beginning, in the middle and/or at the end. A wildcard stands for an arbitrary number of arbitrary characters or nothing. The spelling of the items with upper case and lower case letters and empty spaces must exactly match the spelling of the field names and value names as it is displayed in the module . You can either type in the desired values into the input field, or you can select one or more values from a drop-down list of all available items in the data by pressing the arrow symbol at the right edge of the input field.
  As the first required item group we specify 'LifeInsurance=yes'. That means we look for patterns which have something to do with the fact that a customer has a life insurance contract with the analyzed bank. We enter the text into the editor field of the pop-up dialog and then press **Add**.



- As the second required group we specify `Profession*` and `AccountBalance*`. That means, we enforce that each detected patterns contains an information either on the profession or on the account balance of the customer.



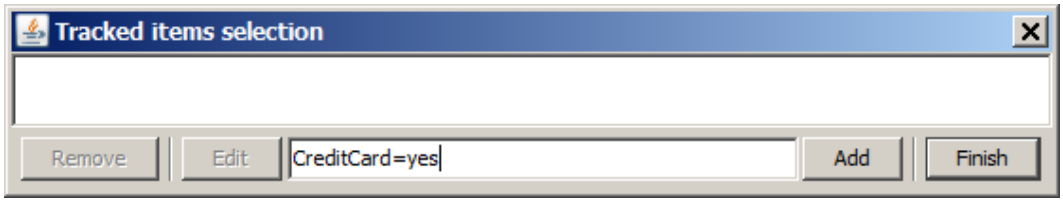
- The buttens at the left of the three 'required item' buttons specify allowed positions of the required items within the association rules to be detected. **Anywhere** means that the item may occur anywhere within the rule, **Rule body** means that the item must occur on the left side ('if') of the rule, **Rule head** means that the item must occur on the right side ('then') of the rule.

- **Suppressed Items** are items which are to be ignored during the pattern search. In our example, we are not interested in any information on joint accounts, therefore we enter `Join*` into the pop-up dialog **suppressed items**.

- If a pair of items or item groups has been specified as **incompatible** (by pairs), then none of the detected associations will contain more than one item out of this set. In the text field of the pop-up dialog, you can enter several patterns, separated by comma (,) without adjacent spaces. If a pattern contains a comma as part of the pattern name, escape it by a backslash (\). Each pattern can contain one or more wildcards (*) at the beginning, in the middle and/or at the end.
  In general it is reasonable to specify items from highly correlated data fields as 'incompatible'. Otherwise one would obtain many patterns with very high lift values in which one item from each of the two highly correlated fields appears. These trivial associations might shadow the truely interesting, non-trivial associations.
  In our example, we define the item pairs `NumberCredits` and `NumberDebits` as well as the pairs `Age` and `DurationClient` as incompatible.

- The **item pair purity** of two items i1 and i2, is the number of transactions in which both items occur divided by the maximum of the absolute supports of the two items. Item pairs with a purity of 1 are 'perfect pairs': whenever i1 occurs in a transaction, also i2 occurs in it, and vice versa.Defining an upper limit for the permitted item pair purity is therefore an alternative means for specifying many single incompatible item pairs. It serves to suppress all trivially highly correlated item pairs from the associations analysis.
  In our example, we have suppressed all item pairs which have a purity of 0.75 (75%) or more.

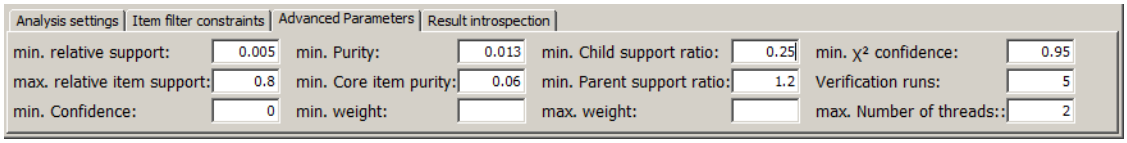
- **Tracked items** are items whose occurrence rate is tracked and shown for every detected association. The tracked rate indicates the probability that the tracked item occurs in a data record or group which supports the current association.
  In our example, we specify that we want to be shown the percentage of credit card users on the support of every single pattern that will be detected.



- **Negative items** are items for which the complement, i.e. the fact that the item does NOT occur, should be treated as a separate item. For example, if the item 'OCCUPATION=Manager' is added to the list of negative items, then the item '!(OCCUPATION=Manager)' is created, and its support is the complement of the support of 'OCCUPATION=Manager'.
  In our example, we specify the item `Profession=inactive` as negative item. That means, we want the fact that a customer has a profession to appear as a new item in the detected patterns.

3.9.6 Advanced pattern statistics constraints

The third tab at the lower end of the screen, **Advanced Parameters**, provides 12 parameters which serve for fine-tuning the detected pattern set based on certain statistical measures.



- The **relative support** of the patterns to be detected in our example must be at least 0.005, or 0.5%. When specifying the parameters for an associations training, you should always specify an lower boundary for the absolute or relative support, otherwise the training can take extremely long time. In our example, however, setting the minimum relative support to 0.005 has no real effect and is redundant since we have already specified a minimum absolute support of 50, and there are about 10000 data groups in the sample data.
- The **relative support of an item** is the item's absolute support divided by the total number of transaction (groups). In other words, the relative support is the a-priori probability that the item occurs in a randomly selected transaction. Items which appear in (almost) every data group often represent trivial information which one does not want to find in the detected patterns. In our example, we have specified an upper boundary of 0.8 in order to suppress items which occur on at least 80% of all transactions.

- The **confidence** of an association rule is the ratio between the rule's support and the rule body's support. An association rule is an association of n items in which n-1 of the n items are considered the 'rule body' and the remaining item is considered the 'rule head'. Hence, n different association rules can be constructed from one association of length n. A rule's confidence is the probability that the rule head is true if one knows for sure that the rule body is true. In our example we have specified that we want to search only those associations in which the confidence of at least one possible split into body and head yields a confidence value of at least 0.2.
- The **purity** of an association is the ratio between the association's support and the support of the most frequent item within the association. In our example, we have specified a minimum required purity of 0.013, that means we accept patterns whose items appear much more frequently than the entire pattern.
- The **core purity** of an association is the ratio between the association's support and the support of the least frequent item within the association. In our example, we have specified a minimum required purity of 0.013, that means we accept patterns in which even the least frequent item appears much more frequently than the entire pattern.
- The **weight** of an association is the mean weight of all data groups which support the association. A minimum or maximum threshold for the associations' weights can only be specified if a weight field has been defined on the input data. Therefore, we leave the two input fields for minimum and maximum weight empty.
- The parameter **minimum child support ratio** defines boundary for the acceptable 'support shrinking rate' when creating expanded associations out of existing associations. An expanded association of n items will be rejected if at least one of the n parent associations has a support which is so large that when multiplied with the minimum shrinking rate, the result is larger than the actual support of the expanded association. In our example we have specified the value of 0.25. That means we suppress the formation of patterns whose support is less than 25% of the support of the least frequent parent pattern.
- **Minimum Parent support ratio** is the acceptable support growth when comparing a given association to its parent associations. A parent association (of n-1 items) will be rejected if its support is less than the support of the current association (of n items) multiplied by the minimum parent support ratio. The effect of this filter criterion is that it reduces the number of detected associations by removing all sub-patterns of long associations whenever the sub-patterns have a support which is not strongly larger than the support of the long association. Inour example, we have set a value of 1.2. That means, parent patterns will be eliminated from the result set whenever their support is less than 120% of the supports of any of their longer child patterns.
- The χ^2 **confidence level** of an association indicates up to which extent each single item is relevant for the association because its occurrence probability together with the other items of the association significantly differs from its overall occurrence

probability.<p> More formally, theχ^2 confidence level is the result of performing n χ^2 tests, one for each item of the association. The null hypothesis for each test is: the occurrence frequency of the item is independent of the occurrence of the item set formed by the other n-1 items.<p> Each of the n tests returns a confidence level (probability) with which the null hypothesis is rejected, and the χ^2 confidence level of the association is set to the minimum of these n rejection confidences.

- **Verification runs** serve to assess whether the detected association or sequential patterns are statistically significant patterns or just random fluctuations (white noise). For each verification run, a separate data base is used. Each data base is generated from the original data by randomly assigning each data field's values to another data row index within the same data field. This approach is called a permutation test. The effect is that correlations and interrelations between different data fields are completely removed from the data.
  If one finds association or sequential patterns on a permuted data base, one can be sure that one has detected nothing but noise. One can record and trace the measure tuples (pattern length, support, lift, purity) of all detected noise patterns. The edge of the resulting point cloud defines the intrinsic 'noise level' of the original data. Patterns detected on the original data can only be considered significant if their corresponding measure triples are well above the noise level.

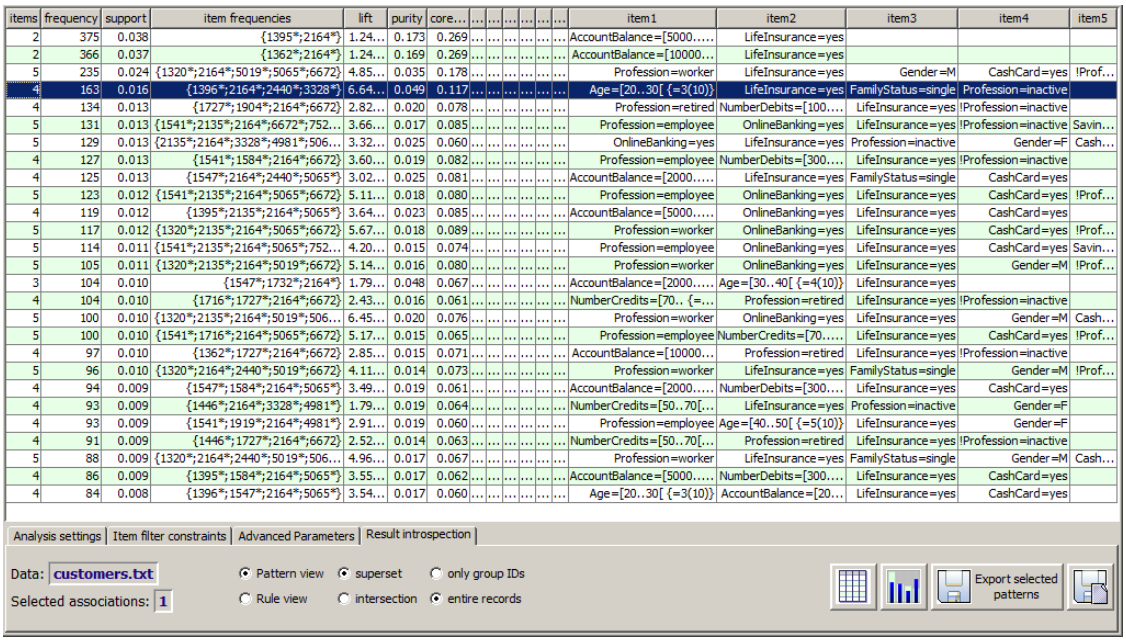
- The parameter **maximum number of threads** specifies an upper limit for the number of parallel threads used for reading and compressing the data. If no number or a number smaller than 1 is given here, the maximum available number of CPU cores will be used in parallel.

3.9.7 Result display options

The fourth tab within the tool bar at the lower border of the associations analysis window offers some capabilites to modify the display mode of the detected associations and to introspect and export them. Some of the buttons only become enabled if you have selected one or more patterns by mouse clicks in the result table above the tool bar.

The screenshot shown below results if one performs the parameter settings described in the previous sections, presses the button **Start training** in the first tab and finally selects one of the resulting patterns by left mouse click.

The tabular view of detected patterns contains the statistical measures of each pattern and its content, the items which form the pattern. The most important statistical measures are, from left to right: the number of items in the pattern, the pattern's absolute and relative support, the absolute supports of the involved items, the lift, purity and core item purity, and finally the list of the items which form the pattern.

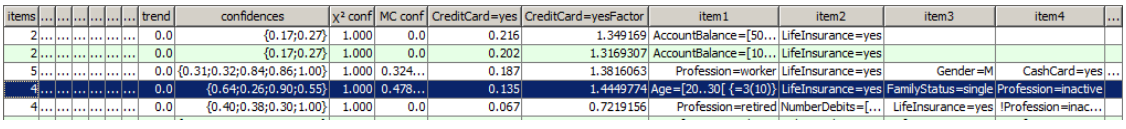


items	frequency	support	item frequencies	lift	purity	core...	item1	item2	item3	item4	item5
2	375	0.038	{1395*;2164*}	1.24...	0.173	0.269	AccountBalance=[5000.....	LifeInsurance=yes			
2	366	0.037	{1362*;2164*}	1.24...	0.169	0.269	AccountBalance=[10000...	LifeInsurance=yes			
5	235	0.024	{1320*;2164*;5019*;5065*;6672}	4.85...	0.035	0.178	Profession=worker	LifeInsurance=yes	Gender=M	CashCard=yes	!Prof...
4	163	0.016	{1396*;2164*;2440*;3328*}	6.64...	0.049	0.117	Age=[20..30[{=3(10)}	LifeInsurance=yes	FamilyStatus=single	Profession=inactive	
4	134	0.013	{1727*;1904*;2164*;6672}	2.82...	0.020	0.078	Profession=retired	NumberDebits=[100....	LifeInsurance=yes	!Profession=inactive	
5	131	0.013	{1541*;2135*;2164*;6672*;752...	3.66...	0.017	0.085	Profession=employee	OnlineBanking=yes	LifeInsurance=yes	!Profession=inactive	Savin...
5	129	0.013	{2135*;2164*;3328*;4981*;506...	3.32...	0.025	0.060	OnlineBanking=yes	LifeInsurance=yes	Profession=inactive	Gender=F	Cash...
4	127	0.013	{1541*;1584*;2164*;6672}	3.60...	0.019	0.082	Profession=employee	NumberDebits=[300....	LifeInsurance=yes	!Profession=inactive	
4	125	0.013	{1547*;2164*;2440*;5065*}	3.02...	0.025	0.081	AccountBalance=[2000.....	LifeInsurance=yes	FamilyStatus=single	CashCard=yes	
5	123	0.012	{1541*;2135*;2164*;5065*;6672}	5.11...	0.018	0.080	Profession=employee	OnlineBanking=yes	LifeInsurance=yes	CashCard=yes	!Prof...
4	119	0.012	{1395*;2135*;2164*;5065*}	3.64...	0.023	0.085	AccountBalance=[5000.....	OnlineBanking=yes	LifeInsurance=yes	CashCard=yes	
5	117	0.012	{1320*;2135*;2164*;5065*;6672}	5.67...	0.018	0.089	Profession=worker	OnlineBanking=yes	LifeInsurance=yes	CashCard=yes	!Prof...
5	114	0.011	{1541*;2135*;2164*;5065*;752...	4.20...	0.015	0.074	Profession=employee	OnlineBanking=yes	LifeInsurance=yes	CashCard=yes	Savin...
5	105	0.011	{1320*;2135*;2164*;5019*;6672}	5.14...	0.016	0.080	Profession=worker	OnlineBanking=yes	LifeInsurance=yes	Gender=M	!Prof...
3	104	0.010	{1547*;1732*;2164*}	1.79...	0.048	0.067	AccountBalance=[2000.....	Age=[30..40[{=4(10)}	LifeInsurance=yes		
4	104	0.010	{1716*;1727*;2164*;6672}	2.43...	0.016	0.061	NumberCredits=[70.. {=...	Profession=retired	LifeInsurance=yes	!Profession=inactive	
5	100	0.010	{1320*;2135*;2164*;5019*;506...	6.45...	0.020	0.076	Profession=worker	OnlineBanking=yes	LifeInsurance=yes	Gender=M	Cash...
5	100	0.010	{1541*;1716*;2164*;5065*;6672}	5.17...	0.015	0.065	Profession=employee	NumberCredits=[70.....	LifeInsurance=yes	CashCard=yes	!Prof...
4	97	0.010	{1362*;1727*;2164*;6672}	2.85...	0.015	0.071	AccountBalance=[10000...	Profession=retired	LifeInsurance=yes	!Profession=inactive	
5	96	0.010	{1320*;2164*;2440*;5019*;6672}	4.11...	0.014	0.073	Profession=worker	LifeInsurance=yes	FamilyStatus=single	Gender=M	!Prof...
4	94	0.009	{1547*;1584*;2164*;5065*}	3.49...	0.019	0.061	AccountBalance=[2000.....	NumberDebits=[300....	LifeInsurance=yes	CashCard=yes	
4	93	0.009	{1446*;2164*;3328*;4981*}	1.79...	0.019	0.064	NumberCredits=[50..70[...	LifeInsurance=yes	Profession=inactive	Gender=F	
4	93	0.009	{1541*;1919*;2164*;4981*}	2.91...	0.019	0.060	Profession=employee	Age=[40..50[{=5(10)}	LifeInsurance=yes	Gender=F	
4	91	0.009	{1446*;1727*;2164*;6672}	2.52...	0.014	0.063	NumberCredits=[50..70[...	Profession=retired	LifeInsurance=yes	!Profession=inactive	
5	88	0.009	{1320*;2164*;2440*;5019*;506...	4.96...	0.017	0.067	Profession=worker	LifeInsurance=yes	FamilyStatus=single	Gender=M	Cash...
4	86	0.009	{1395*;1584*;2164*;5065*}	3.55...	0.017	0.062	AccountBalance=[5000.....	NumberDebits=[300....	LifeInsurance=yes	CashCard=yes	
4	84	0.008	{1396*;1547*;2164*;5065*}	3.54...	0.017	0.060	Age=[20..30[{=3(10)}	AccountBalance=[20...	LifeInsurance=yes	CashCard=yes	

The items describing numeric data field values contain, in addition to the value range limits, an extra information within curly braces: the position of the value range within the overall value distribution of the numeric data field. For example, the item `Age=[20..30[ \=3(10)\` means that the age range from 20 (incl.) to 30 (excl.) is the third smallest out of 10 value ranges, hence the age value is below average but not strongly below average.

The numbers in the table column **item frequencies** contain the absolute supports of the different items of the pattern, in the same order in which the item names appear in the columns at the right end of the result table. If the number is marked by a star (*), the corresponding item belongs to the core of the pattern. That means that each partial pattern in which this item has been removed has a larger support than the original pattern.

The tabular result view also contains some more advanced information on the detected patterns. In the figure shown below these columns have been enlarged and thus highlighted:



items	trend	confidences	χ^2 conf	MC conf	CreditCard=yes	CreditCard=yesFactor	item1	item2	item3	item4
2	0.0	{0.17;0.27}	1.000	0.0	0.216	1.349169	AccountBalance=[50...	LifeInsurance=yes		
2	0.0	{0.17;0.27}	1.000	0.0	0.202	1.3169307	AccountBalance=[10...	LifeInsurance=yes		
5	0.0	{0.31;0.32;0.84;0.86;1.00}	1.000	0.324...	0.187	1.3816063	Profession=worker	LifeInsurance=yes	Gender=M	CashCard=yes
4	0.0	{0.64;0.26;0.90;0.55}	1.000	0.478...	0.135	1.4449774	Age=[20..30[{=3(10)}	LifeInsurance=yes	FamilyStatus=single	Profession=inactive
4	0.0	{0.40;0.38;0.30;1.00}	1.000	0.0	0.067	0.7219156	Profession=retired	NumberDebits=[...	LifeInsurance=yes	!Profession=inac...

- The measure **trend** indicates whether an association pattern has become more important recently (value > 0) or less important (value < 0). The measure can only be computed if an order field (time stamp field) has been defined on the input data. If an oder field exists, the trend number is calculated from a histogram of the order field as it is displayed in the module Multivariate Exploration. This is done by comparing the value distribution of the order field on the data groups which support the given pattern to the corresponding value distribution on the entire data.More familiarly spoken: if the blue bars are higher than the light green bars on

the right side of the histogram, the trend value is positive, if they are smaller than the light green bars, the value is negative. More precisely, the displayed trend value is computed out of three contributions: a 'long-term' point of view which compares the two series of bars on all N available order field value ranges, a 'short-term' point of view which compares the two series on the last 5 available data points, and a 'mid-term' point of view which compares the two series on the last M time points, where M is the geometrical mean of 5 and N.More familiarly spoken: if the blue bars are higher than the light green bars on the right side of the histogram, the trend value is positive, if they are smaller than the light green bars, the value is negative. More precisely, the displayed trend value is computed out of three contributions: a 'long-term' point of view which compares the two series of bars on all N available order field value ranges, a 'short-term' point of view which compares the two series on the last 5 available data points, and a 'mid-term' point of view which compares the two series on the last M time points, where M is the geometrical mean of 5 and N.

- The measure **weight** contains the weight of the association pattern. The measure is only displayed if a weight field has been defined on the data. If this is the case, the measure contains the mean weight of all transactions (data groups) on which the pattern occurs.

- The **confidence** numbers display the n different confidences of the n possible association rules that can be formed out of the association pattern of n items by interpreting one item as the rule head (right side) and n-1 items as the rule body (left side). The i-th confidence value corresponds to the rule in which the i-th item is the head item.

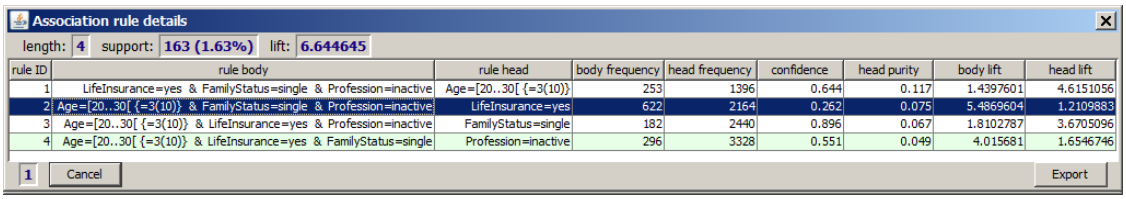
- The measure χ^2 **confidence** displays the result of the χ^2 significance test described in section Advanced parameters. The last section of this chapter explains how this number can be interpreted.

- The measure **MC confidence** (Monte Carlo confidence) is only displayed if verification runs have been performed (see section Advanced parameters). The last section of this chapter explains how this number can be interpreted.

- Für each **tracked item** specified on the item filters tab of the tool bar, the result table contains two columns: one column labeled with the name of the item, in our example `Creditcard=yes`, the second one labeled with the name of the item plus `Factor`, in our example `Creditcard=yesFactor`. The first column value displays the fraction of data groups which contain the tracked item within the data groups on which the current pattern occurs. Hence, the value indicates whether the tracked item occurs more or less frequently on the supporting data groups of the pattern compared to the overall data.
  The second column indicates whether the appearance of the entire current pattern has an impact on the occurrence rate of the tracked item which exceeds the effect that the pattern's single items have on the occurrence of the tracked item. Let us look at the blue table row in the picture shown above. It contains the value 1.44. That means, the percentage of credit card users is 44% higher on the supporting

data groups of the pattern than the geometrical mean of the 4 percentages of credict card users on the 4 sets of data groups on which the tracked item occurs with one of the 4 items of the pattern. Hence, the coincidence of all 4 items of the patterns seems to have an increasing effect on credit card usage.

Clicking on a table row with the right mouse button opens a detail view of the association in a separate pop-up window. The detail view displays the n different possibilities to interpret the association as an association rule with exactly one item as the rule head. For each rule, the detail view contains the absolute support of the rule body and the rule head, the rule's confidence, the lift of the rule body pattern and the rule lift.

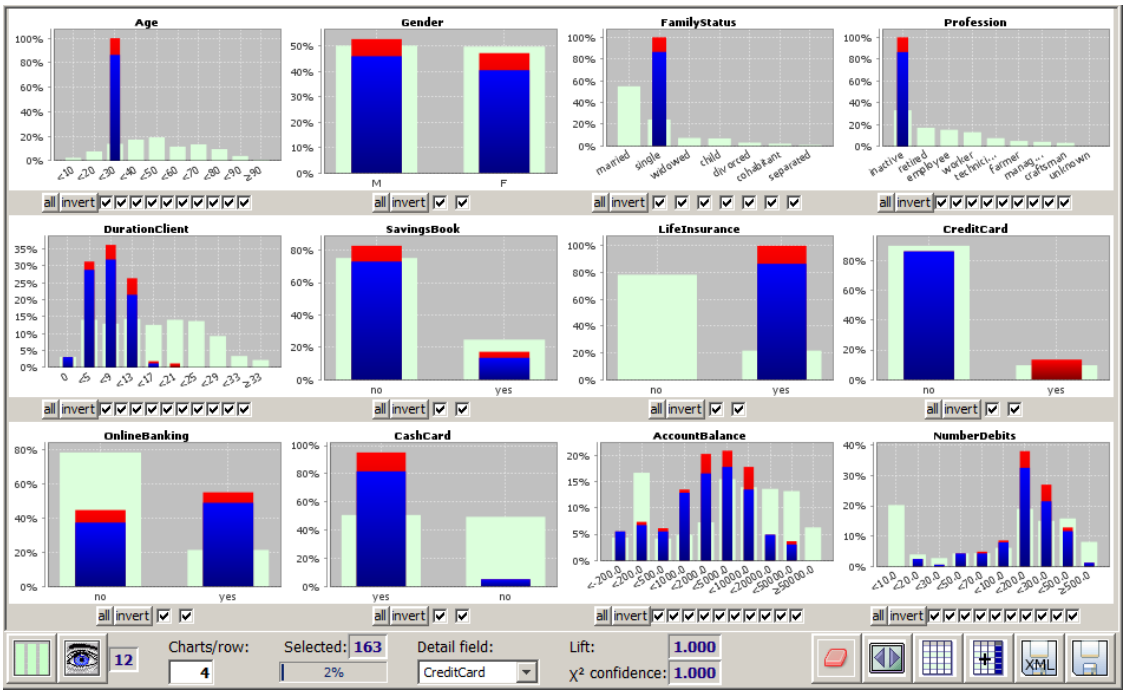

Association rule details
length: 4 support: 163 (1.63%) lift: 6.644645
rule ID | rule body | rule head | body frequency | head frequency | confidence | head purity | body lift | head lift
1 | LifeInsurance=yes & FamilyStatus=single & Profession=inactive | Age=[20..30[{=3(10)} | 253 | 1396 | 0.644 | 0.117 | 1.4397601 | 4.6151056
2 | Age=[20..30[{=3(10)} & FamilyStatus=single & Profession=inactive | LifeInsurance=yes | 622 | 2164 | 0.262 | 0.075 | 5.4869604 | 1.2109883
3 | Age=[20..30[{=3(10)} & LifeInsurance=yes & Profession=inactive | FamilyStatus=single | 182 | 2440 | 0.896 | 0.067 | 1.8102787 | 3.6705096
4 | Age=[20..30[{=3(10)} & LifeInsurance=yes & FamilyStatus=single | Profession=inactive | 296 | 3328 | 0.551 | 0.049 | 4.015681 | 1.6546746
1
Cancel
Export

In the tool bar tab **Result introspection** the following options are available:

- The information displayed at the left end of the tab contains the name of the data source and the number of patterns which are currently selected. When at least one pattern is selected, a mouse-click on the label **Selected associations** creates a pop-up window in which a SQL SELECT statement is displayed which corresponds to the currently selected patterns.
- Next to this information there are two vertically positioned radio buttons with which you can switch between the default 'pattern view' and an an alternative 'rule view' in the result table displayed above. In the default display mode, each detected association pattern is represented by one single table row. In the alternative 'rule view', each pattern of n items is displayed in the form of n association rules, each one with another head item. Hence, the second variant is more complex but contains a lot of additional information for all the displayed rules. For displaying this information in the default pattern view, you would have to right-click on every single table row in order to display the row's detail view.
- The next vertical pair of radio buttons determines what happens if several associations have been selectend and then the button is pressed. The button's purpose is to display those data groups which support the selected associations. The question is, does this mean the intersection or the superset of the supports of the single selected patterns? This question is answered by the choice made in these radio buttons.
- The rightmost vertical pair of radio buttons has a similar function to the pair next to it: it specifies whether pressing the button dispays entire data sets or only the data record numbers or data group IDs of the data groups which support the selected patterns.

- The button opens an additional window which shows the data groups on which the currently selected association patterns occur. Whether the new window contains full-width data records or only record or group IDs, and whether it contains the intersection or the superset of the data groups supporting the single selected patterns, is defined by the radio buttons described above.

- The button opens an additional window in which the data groups on which the currently selected patterns occur can be visually explored. Whether the new window contains the intersection or the superset of the data groups supporting the single selected patterns, is defined by the radio buttons described above.
  The new window provides the entire functionality of the module multivariate analysis. The screenshot shown below explores the data groups which support the pattern of length 4 which has been taken as an example in the previously shown pictures. Then we have chosen the data field `Creditcard`as detail structure field. Now the blue and red bars are indicating how the non-credit-card-users and the credit card users behave within the customer group which support the selected pattern.



- Using the button you can export the currently selected patterns, or all patterns if none has been selected, into a <TAB> separated flat text file, into a PMML `AssociationModel` or into a series of SQL SELECT statements.

- Using the button you can export the data groups supporting the currently selected patterns into a <TAB> separated flat text file or into a spreadsheet in `.xlsx` format.

3.9.8 Pattern verification and significance assurance

At the end of the chapter on associations analysis we want to discuss how one can make sure that a detected pattern is a statistically significant pattern and not just a random statistical fluctuation, so called white noise, in the data. This issue is often completely left aside in traditional books on data mining and in many existing software packages.

Synop Analyzer provides two means for targeting this issue: one can calculate a so-called χ^2 confidence level for each pattern, and one can perform one or more verification runs on artificially permuted versions of the original data which serve to define the so-called noise level and the associated 'Monte Carlo confidence' that the given pattern's statistical key measures exceed tht noise level, making it a significant pattern. In this section, the two confidence measures and their interpretation shall be discussed in detail.

As an example, let us look at one concrete association pattern which we have taken as an example several times in this chapter:

items	...	...	...	...	...	...	trend	confidences	χ² conf	MC conf	CreditCard=yes	CreditCard=yesFactor	item1	item2	item3	item4	...
2	...	...	...	...	...	...	0.0	{0.17;0.27}	1.000	0.0	0.216	1.349169	AccountBalance=[50...	LifeInsurance=yes			
2	...	...	...	...	...	...	0.0	{0.17;0.27}	1.000	0.0	0.202	1.3169307	AccountBalance=[10...	LifeInsurance=yes			
5	...	...	...	...	...	...	0.0	{0.31;0.32;0.84;0.86;1.00}	1.000	0.324...	0.187	1.3816063	Profession=worker	LifeInsurance=yes	Gender=M	CashCard=yes	...
4	...	...	...	...	...	...	0.0	{0.64;0.26;0.90;0.55}	1.000	0.478...	0.135	1.4449774	Age=[20..30[{=3(10)}	LifeInsurance=yes	FamilyStatus=single	Profession=inactive	
4	...	...	...	...	...	...	0.0	{0.40;0.38;0.30;1.00}	1.000	0.0	0.067	0.7219156	Profession=retired	NumberDebits=[...	LifeInsurance=yes	!Profession=inac...	

The highlighted sample pattern has length 4, absolute support (frequency) 163, relative support of 1.6%, a lift value of 6.64, the χ^2 confidence of 1.000 and the Monte Carlo confidence of 0.58. What does that mean for the significance of the pattern, and why is the χ^2 confidence of this pattern (and of most other patterns) much larger than the Monte Carlo confidence?

For answering these questions we start with remembering the definition of χ^2 confidence. A pattern of n items with absolute support S has a χ^2 confidence of x% if for each of the n items, the following holds: the appearance probability of the item in the presence of the n-1 other items of the pattern differs so strongly from the a-priori appearance probability of the item on the overall data that this difference is in x out of 100 cases greater than the difference in appearance probilities which results from comparing a randomly selected subset of S data groups to the entire data. More familiarly spoken, that means roughly the following: x out of 100 association patterns which do **not** represent a statistically significant relation on the data and which have the same pattern length and support as the given pattern, would have a lift value closer to 1 than the given pattern. Inversely, this also means: even if a pattern has a χ^2 confidence value of 0.9999, 1 out of 10000 randomly chosen noise patterns of the same length and support would have a lift value as strong as the given pattern.A typical associations analysis - if not almost all items appearing in the detected patterns have been specified as 'required items' by the user - examines billions or even trillions of candidate patterns. Therefore, it is highly probable that e few random noise patterns make it into the displayed result which have a χ^2 confidence of 0.999 or even 1.000.

In summary we can conclude: that a pattern has a χ^2 confidence of 0.95 or higher is a necessary but not a sufficient condition for the pattern's statistical significance. The

condition is only sufficient if the search space of candidate patterns during the analysis was very small, that means if only a few patterns were evaluated. In all other cases, one needs other significance measures for finally classifying a pattern as significant or not.

In these latter cases, the Monte Carlo confidence level, which is based on verification runs and permutation tests, gives a more reliable significance estimation. The method first calculates a 'maximum noise level' for each pair of (pattern length, support) based on all available verification runs. The maximum noise level takes into account all recorded lift, purity and core item purity values of the detected patterns on the verification data. From each triple (lift, purity, core item purity), a number NL(length,support) is calculated, and the maximum noise level MNL is the maximum of all recorded NL(length,support). For pairs (length,support) for which not enough patterns have been found within the verification runs, the maximum noise level is interpolated and estimated from neigbored MNL values.Once the MNLs have been established, we calculate the corresponding quality number Q as a function of lift, purity and core item purity for each detected pattern on the real data and compare it to the MNL for the same length, support, lift, purity and core item purity. The Monte Carlo confidence is a function of Q minus MNL which is calibrated such that the result is 0.45 if Q equals MNL and 0.95 if Q equals 1.5 MNL.

Familiarly spoken, we can interpret the Monte Carlo confidence as follows: a value of about 0.5 means that on all verification runs not a single fluctuation pattern has been found with the same combined significance of the values pattern length, support, lift, purity and core item purity as the current pattern. This is a good evidence for the fact that the current pattern is statistically significant. The evidence becomes even stronger if the MC confidence goes towards 1.0. That means, our sample pattern, which has MC conf=0.58, is with high probability statistically significant, whereas the pattern below our example pattern in the result table could be random noise, even though its χ^2 confidence is 1.000.

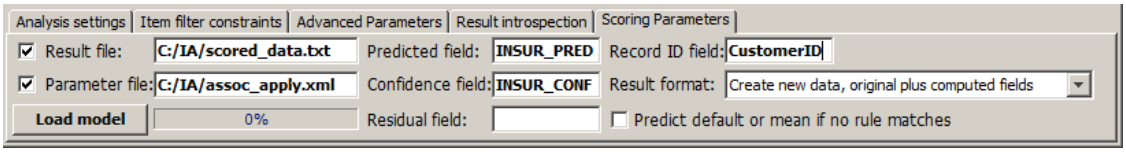
3.9.9 Applying association models to new data ('Scoring')

Association models can be applied to new data in order to create predictions on these data. For example, an associations model could use the click history of a web shop user to decide which product offers are to be shown to this user. Another associations model could serve as an early warning system in a production process, predicting upcoming problems and faulty products. A third associations model could classify credit demands into a high risk and a low risk group. This application of associations models to new data for predictive purposes is called 'scoring'.

In the current version of Synop Analyzer, associations models must satisfy a certain precondition for being usable for scoring: all association rules in the model must have rule heads ('then' sides) containing values of one single data field. This data field is called the target field of the model. In the three sample applications cited above (web shop, production monitoring, credit risk), the target fields could be `ARTICLE`, `ERROR` and `RISK_CLASS`.

If all rules of the model only contain information ('items') from one single data field, the precondition for scoring is trivially satisfied. If not, you can enforce the precondition by defining one or more required items of type **Rule head** when training the model. In this case, you must make sure all required head items are values or value ranges of one single data field.

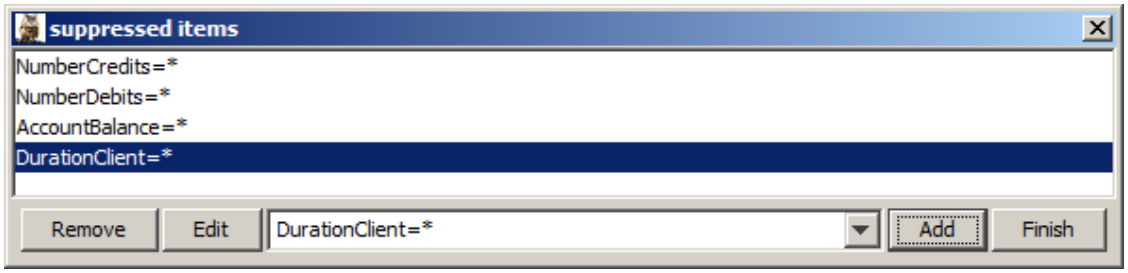
You load and apply an associations model by first opening and reading the new data, by then pressing the button in order to start the associations analysis module and by then clicking the button **Load model** in the tab **Scoring Settings** of the tool bar at the lower end of the panel's GUI window.



In the following sections we will demonstrate the process of associations scoring with the help of a concrete example use case: using an associations model we want to predict the propensity of newly acquired bank customers to sign a life insurance contract.

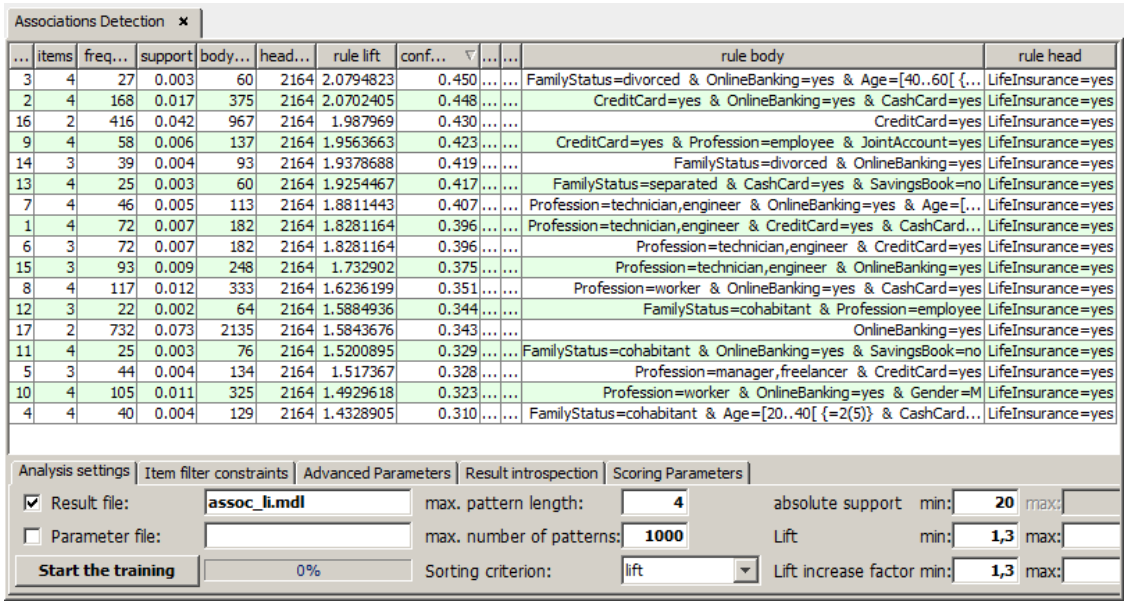
For this purpose, we load the sample data `doc/sample_data/customers.txt`. We keep the default data import settings with one exception: the number of bins for numeric fields (**Bins:**) is reduced from 10 to 5. Then we start the associations analysis module and train a model called `assoc_li.mdl`, using the following parameter settings:

- Required item `LifeInsurance=yes` of type **Rule head**,
- Incompatible items `FamilyStatus=*` and `JointAccount=*` (because these two fields are highly correlated),
- Suppressed items `NumberCredits=*`, `NumberDebits=*`, `AccountBalance=*` and `DurationClient` (because the information on accounting activity and acount balance are not reliably available for new customers and the duration of the business relationship is always 0)



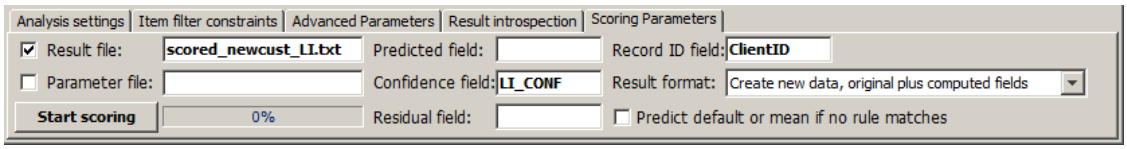
- Minimum absolute support 20, minimum lift 1.3, minimum lift increase factor 1.3.

The model trained with these settings contains 17 rules. The strongest rule predicts a probability of 45% that a customer with the properties given on the left side of the rule will sign a life insurance contract.


Associations Detection
... items freq... support body... head... rule lift conf... ... ... rule body rule head
3 4 27 0.003 60 2164 2.0794823 0.450 ... ... FamilyStatus=divorced & OnlineBanking=yes & Age=[40..60[{... LifeInsurance=yes
2 4 168 0.017 375 2164 2.0702405 0.448 ... ... CreditCard=yes & OnlineBanking=yes & CashCard=yes LifeInsurance=yes
16 2 416 0.042 967 2164 1.987969 0.430 ... ... CreditCard=yes LifeInsurance=yes
9 4 58 0.006 137 2164 1.9563663 0.423 ... ... CreditCard=yes & Profession=employee & JointAccount=yes LifeInsurance=yes
14 3 39 0.004 93 2164 1.9378688 0.419 ... ... FamilyStatus=divorced & OnlineBanking=yes LifeInsurance=yes
13 4 25 0.003 60 2164 1.9254467 0.417 ... ... FamilyStatus=separated & CashCard=yes & SavingsBook=no LifeInsurance=yes
7 4 46 0.005 113 2164 1.8811443 0.407 ... ... Profession=technician,engineer & OnlineBanking=yes & Age=[... LifeInsurance=yes
1 4 72 0.007 182 2164 1.8281164 0.396 ... ... Profession=technician,engineer & CreditCard=yes & CashCard... LifeInsurance=yes
6 3 72 0.007 182 2164 1.8281164 0.396 ... ... Profession=technician,engineer & CreditCard=yes LifeInsurance=yes
15 3 93 0.009 248 2164 1.732902 0.375 ... ... Profession=technician,engineer & OnlineBanking=yes LifeInsurance=yes
8 4 117 0.012 333 2164 1.6236199 0.351 ... ... Profession=worker & OnlineBanking=yes & CashCard=yes LifeInsurance=yes
12 3 22 0.002 64 2164 1.5884936 0.344 ... ... FamilyStatus=cohabitant & Profession=employee LifeInsurance=yes
17 2 732 0.073 2135 2164 1.5843676 0.343 ... ... OnlineBanking=yes LifeInsurance=yes
11 4 25 0.003 76 2164 1.5200895 0.329 ... ... FamilyStatus=cohabitant & OnlineBanking=yes & SavingsBook=no LifeInsurance=yes
5 3 44 0.004 134 2164 1.517367 0.328 ... ... Profession=manager,freelancer & CreditCard=yes LifeInsurance=yes
10 4 105 0.011 325 2164 1.4929618 0.323 ... ... Profession=worker & OnlineBanking=yes & Gender=M LifeInsurance=yes
4 4 40 0.004 129 2164 1.4328905 0.310 ... ... FamilyStatus=cohabitant & Age=[20..40[{=2(5)} & CashCard... LifeInsurance=yes
Analysis settings
Item filter constraints
Advanced Parameters
Result introspection
Scoring Parameters
Result file: assoc_li.mdl
Parameter file:
Start the training
0%
max. pattern length: 4
max. number of patterns: 1000
Sorting criterion: lift
absolute support min: 20 max:
Lift min: 1,3 max:
Lift increase factor min: 1,3 max:

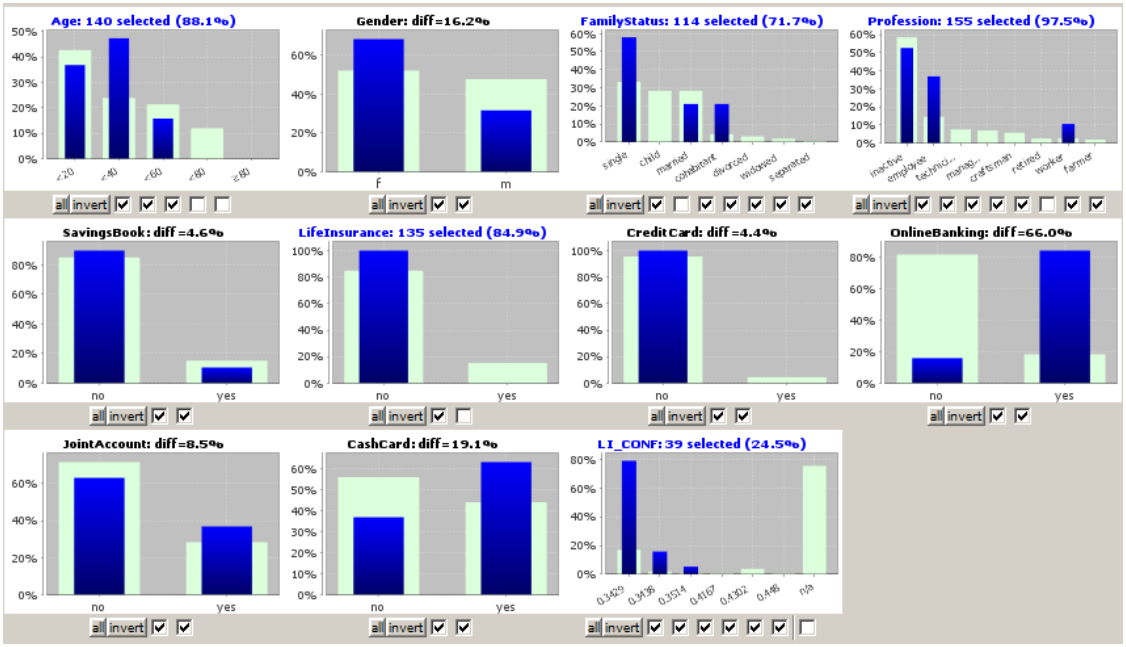
Now we want to use the generated model for predicting the propensity of 159 new customers for signing life insurance contracts. The new customers' data reside in the file `newcustomers_159.txt`. We load these data as a new Synop Analyzer data source. The value range discretizations of the numeric fields of the new data must be identical to the range discretizations that were in place when the model was created. In our case, we use the pop-up window **Settings → Field discretizations** to make sure the field `Age` has the range boundaries 20, 40, 60 and 80. For the field `ClientID` we specify the usage type **group** in the dialog **Active fields**.

On this in-memory data source we start the associations analysis module and move to the tab **Scoring Parameters** in the tool bar at the lower end of the screen. Here, we enter the name of the file in which the scoring results are to be stored (`scored_newcust_LI.txt`), we define the scoring result data fields to be contained in that file and we specify that the new file should be a copy of the existing file `newcustomers_159.txt` plus the new computed data fields. (**Create new data, original plus computed fields**).Since all association rules in our model predict the same value (`LifeInsurance=yes`), we do not need a new data field **Predicted field**. Instead, we are interested in the predicted probability of that value, therefore we define a **Confidence field** and call it `LI_CONF`. For being able to identify the single customers in the new data, we make sure the key field `ClientID` is contained in the new data and serves as **Record ID field**.



By means of the button **Start scoring** we create the scoring results, write the desired result file to disk and open the resulting data as a new in-memory data source in Synop Analyzer, that means as a new tab in the left column of the Synop Analyzer workbench.

We introspect the scoring result data with the module 'multivariate exploration'. We see that the model has created a non-empty propensity probability for 39 of the 159 new customers. But some of these 39 customers should be filtered out because they already have a life insurance, they have an age of 60 or more years or because they are children or retired persons. There remain 19 new customers which are interesting for selling life insurance contracts:



Via the button we submit the selected 19 data records to a last visual examination. Then we can use the button **Export** to save the resulting list to a flat file or Excel spreadsheet, or we can use the main menu button **Report** to create a HTML or PDF report.

Multivariate Exploration | Data Subset

ClientID	Age	Gender	FamilyStatus	Profession	SavingsBook	LifeInsura...	CreditCard	OnlineBan...	JointAccount	CashCard	LI_C...
P0196848	20	m	single	worker	no	no	no	yes	no	yes	0.3514
P0195197	31	f	cohabitant	employee	no	no	no	no	yes	no	0.3438
P0195241	26	f	cohabitant	employee	yes	no	no	no	yes	no	0.3438
P0195778	47	f	cohabitant	employee	yes	no	no	no	yes	no	0.3438
P0195682	18	f	single	inactive	no	no	no	yes	no	yes	0.3429
P0195686	17	m	single	inactive	no	no	no	yes	no	yes	0.3429
P0195839	18	m	single	inactive	no	no	no	yes	no	yes	0.3429
P0195969	17	m	single	inactive	no	no	no	yes	no	yes	0.3429
P0196161	30	f	single	inactive	no	no	no	yes	no	yes	0.3429
P0196499	18	f	single	inactive	no	no	no	yes	no	yes	0.3429
P0196538	16	m	single	inactive	no	no	no	yes	no	yes	0.3429
P0196713	44	m	single	employee	no	no	no	yes	no	no	0.3429
P0196804	29	f	married	employee	no	no	no	yes	yes	no	0.3429
P0197007	20	f	single	inactive	no	no	no	yes	no	yes	0.3429
P0197062	19	f	single	inactive	no	no	no	yes	no	yes	0.3429
P0197067	23	f	married	worker	no	no	no	yes	yes	no	0.3429
P0197085	25	f	cohabitant	inactive	no	no	no	yes	no	yes	0.3429
P0197150	26	f	married	employee	no	no	no	yes	yes	no	0.3429
P0197536	43	f	married	employee	no	no	no	yes	yes	yes	0.3429

Number of groups: 19 Column width (in pixels): 75 Export

3.10 The Sequential Patterns Analysis module

3.10.1 Introduction to Sequential Patterns Analysis

Sequential patterns analysis is a variant of associations analysiswhich is suitable for data containing a time stamp or a more general data field with ordering information.

Within Synop Analyzer, the sequential patterns analysis module is started using the button

in the left screen column. The button is only active on input data on which an 'entity' field, an 'order' field and a 'group' field have been defined. The Group field and the Order field can be identical. In this case, duplicate the data field in the active fields dialog and specify the original data field as the group field and the duplicated field as the order field.

The result of a sequential patterns analysis is a sequences model, that means a collection of sequential patterns which have been detected during the sequences training run on the training data set. The model can be applied to a new data source in a so-called sequences scoring step. In Synop Analyzer's sequential patterns analysis panel, you can visualize and introspect the sequences model in tabular form, sort, filter and export the filtered results to flat files or into the inter-vendor standard XML format PMML. Furthermore, you can explore and export the support of selected sequential patterns, that means the data sets on which the selected patterns occur.

In the following sections, we will refer to many notations and concepts which have been introduced and explained in the documentation chapter on associations analysis, in particular in the section Definitions and notations of that chapter. Therefore, we recommend to read that chapter and to become familiar with the concepts of associations analysis before starting to use the sequential patterns analysis module.

Unlike an association pattern, a sequential pattern or sequence is a time-ordered combination of several sets of items, a so-called sequence of item sets, in which the items within each item set occur at the same time and consequtive item sets are separated by time steps larger than zero.

An example for a sequence is the following one, based on supermarket purchase data:
`(diapers size 1 (new born) & baby cleansing tissues)` →`[4±1 months]`→ `baby food 4th-6th month`
The sequence consists of two item sets and contains the fact the a certain group of supermarket customers starts buying `diapers size 1` and `soft baby cleansing tissues` at a certain point of time, and the same customers often start buying baby food for 4 to 6 months old babies 4 months plus/minus one month after buying their first diapers and baby tissues.

A sequence rule is a sequence in which the last time step is interpreted as the separation between the rule body (left hand side) and the rule head (right hand side).

The table below lists typical use cases for sequential patterns analysis [Ballard, Rollins, Dorneich et al., Dynamic Warehousing: Data Mining made easy]:

industry	**use case**	**entity field**	**group field**	**typical body item**	**typical head item**
retail	upselling analysis	customer ID	bill ID or purchase ID	a purchased article	another purchased article
manufacturing	quality assurance	product (e.g. vehicle ID)	process step or timestamp	component, production condition	problem, error ID
medicine	medical study evaluation	patient or test person	treatment step or date	single treatment info	medical impact

3.10.2 Input data formats

As mentioned in the first section of this chapter, each data source on which a sequential patterns analysis is to be performed must contain a so-called entity field and an order or timestamp field. These fields must have been declared in the active fields dialog of the input data panel. The entity field contains the subjects (entities) on which time-ordered patterns habe been observed, e.g. customers, vehicles, or patients.

Another required property of the data is that they are sorted by entity field values and, if available, by group field values. If the data are read from a database, Synop Analyzer automatically assures that property by issuing a `SELECT` statement with an appropriate `ORDER BY` clause. If the data are read from flat file or from a spreadsheet, the user is responsible for bringing the data into the correct order. Synop Analyzer will issue a warning message if the data are not correctly ordered.

If these prerequisites are fulfilled, Synop Analyzer's sequential patterns analysis module is prepared for working with three different data formats:

- **The transactional or pivoted data format:**
  Often, the input data for sequences analysis are available in a format in which one column is the so-called group field and contains transaction IDs, one or more additional fields are the so-called item fields and contain items, i.e. the information on which associations are to be detected.
  The file `doc/sample_data/RETAIL_PURCHASES.txt` is an example for such a data format: the field `PURCHASE_ID` is the group field, the field `ARTICLE` contains the real information, namely the IDs of the purchased articles; the field `CUSTOMER_ID` is the entity field and `DATE` the order field.
  In the transactional data format, the items appearing in the detected sequential patterns are a combination of field name and field value if there is more than one item field; the name of the item field is omitted if all items come from one single field (such as the field `ARTICLE`).

- **The data format with Boolean fields:**
  You can also detect sequential patterns on input data which do not have a group field (that means each data row represents a separate transaction) and in which each single 'item', i.e. each single event or fact, has its own two-valued (Boolean) data field which indicates whether or not the item occurs in the transaction.
  If the field `PURCHASE_ID` was missing in the sample data `doc/sample_data/RETAIL_PURCHASES.txt` and if there was a separate data field for each existing article ID which contained either 0 or 1, depending on whether or not the corresponding article was purchased in transaction represented by the current data row, then the data would have the data format with Boolean fields.
  If Synop Analyzer detects a data format with Boolean fields, it interprets all Boolean field values starting with '0', '-', 'F' or 'f' (such as 'false'), 'N' or 'n' (such as 'no' or 'n/a') as indicators for 'item does not occur in the transaction', all other values are interpreted as 'item occurs in the transaction'.
  In the data format with Boolean fields the items appearing in the detected patterns contain only the names of the Boolean fields, but not the field values such as 'YES' or '1'.

- **The 'normal' or broad data format:**
  Of course, Synop Analyzer can also detect sequential patterns on 'normal' data in which each single data row is considered one data group and in which there are different data fields of various types which contain the items.
  On these data, the items appearing in the detected patterns always have the form 'field_name=field_ value'.

A general rule, which is valid on all data formats, is: the items which form the detected sequences can only come from active data fields which have not been marked as 'group', 'entity', 'oder' or 'weight'. 'entity' and 'group' field values serve to define data groups covering more than one data row, information from 'order' fields is used to attach a time stamp to each item, and information from 'weight' fields is used to calculate pattern weight coefficients.

3.10.3 Definitions and notations

A sequence or sequence rule can be characterized by the following properties: [Ballard, Rollins, Dorneich et al., Dynamic Warehousing: Data Mining made easy]

- The **items**
  which are contained in the rule body, in the rule head, or in the entire rule.

- **Categories** of the contained items.
  Often, an additional hierarchy or taxonomy for the items is known. For example, the items 'milk' and 'baby food' might belong to the category 'food', 'diapers' might belong to the category 'non-food'. 'axles=17-B' and 'engine=AX-Turbo 2.3' might be members of the category 'component', 'production_chain=3' of category 'production condition', and 'delay > 2 hours' of category 'error state'. Hence, the second

sample rule can be characterized by the fact that its body contains components or production conditions, and its head an error state.

- The **support** of the pattern:
  Absolute support `S` is defined as the total number of entities for which the rule holds. Support or relative support `s` is the fraction of all entities for which the rule holds. Note that this is different from the definition of support of an association pattern, which is defined in terms of data groups (transactions), not in terms of entities.

- The **confidence** of the pattern when interpreted as a rule:
  Confidence `C` is defined as
  `C := s(body →[dt]→ head) / s(body)`.

- The **lift** of the pattern:
  The Lift `L` of a pattern $\texttt{Itemset}_1 \rightarrow[\texttt{dt}_1]\rightarrow \ldots \rightarrow[\texttt{dt}_{n-1}] \texttt{Itemset}_n$ is defined as
  $\texttt{L := s(Itemset}_1 \rightarrow[\texttt{dt}_1]\rightarrow \ldots \rightarrow[\texttt{dt}_{n-1}]\rightarrow \texttt{Itemset}_n\texttt{) / (s(Itemset}_1\texttt{) * ... * s(Itemset}_n\texttt{))}$.
  When interpreting the lift value of a sequence, we can not simply formulate in analogy to what we have done for association patterns: lift > 1 (< 1) means that the pattern appears more (less) frequently than expected assuming that all involved items are statistically independent. The problem is that in the enumerator part of the lift formula given above, we do not count all common occurrences of all involved items but only the occurrences in the correct time order.Therefore, an interpretation of lift values is difficult. One can, however, say that a lift value greater than 0.5 always stands for a positive correlation of the involved items in the given time ordering. Apart from that, lift values should only be used for comparisons (this sequence is more positively correlated than that sequence'), and these comparisons should only be drawn between sequences of the same number of items and the same number of time steps.

- The **purity** of the sequence pattern:
  The purity `P` of a sequence $\texttt{Itemset}_1 \rightarrow[\texttt{dt}_1]\rightarrow \ldots \rightarrow[\texttt{dt}_{n-1}]\rightarrow \texttt{Itemset}_n$ is defined as
  $\texttt{P := s(Itemset}_1 \rightarrow[\texttt{dt}_1]\rightarrow \ldots \rightarrow[\texttt{dt}_{n-1}]\rightarrow \texttt{Itemset}_n\texttt{) / max}_{i=1\ldots n}\texttt{(s(Itemset}_i\texttt{))}$.
  `P` $= 1$ means that the pattern describes a 'perfect sequence': none of the parts $\texttt{Itemset}_i$ ever occurs on any entity without all the other parts in the time ordering defined by the sequence.

- The **weight (cost,price)** of the pattern:
  If a weight field has been defined on the input data, we can calculate the weight of a sequence as the average of summed weights of the entities which support the sequence.

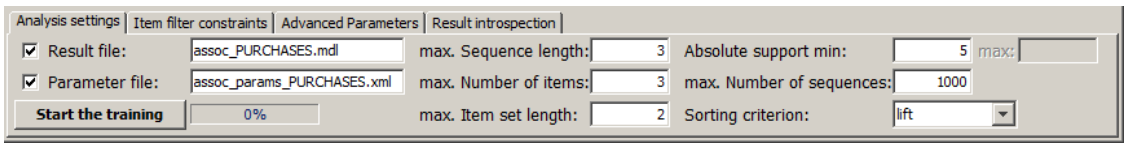
3.10.4 Basic parameters for an Sequential patterns analysis

In Synop Analyzer, an sequence analysis is started by loading a data source - the so-called training data - into memory and by clicking on the button

in the input data panel on the left side of the Synop Analyzer GUI. The button opens a panel named **Sequences Detection**. In the lower part of this panel, you can specify the settings for an sequential patterns analysis and start the search. The detection process itself can be a long-running task, therefore it is executed asynchronically in several parallelized background threads. In the upper part of the panel, the detected sequences - the so-called sequence model - are displayed.

The following paragraphs and screenshots demonstrate the handling of the various sub-panels and buttons at hand of the sample data `doc/sample_data/RETAIL_PURCHASES.txt`. We assume that these data have been imported into Synop Analyzer as described in Name mappings and Taxonomies, that means with `PURCHASE_ID` as group field, `CUSTOMER_ID` as entity field, `DATE` as order field, `PRICE` as weight field and with `doc/sample_data/RETAIL_NAMES_DE_EN.txt` as article names and `doc/sample_data/RETAIL_ARTICLEGROUPS.txt` as article hierarchies.

The first visible tab in the toolbar at the lower end of the screen contains the most important parameters for sequential patterns analysis.



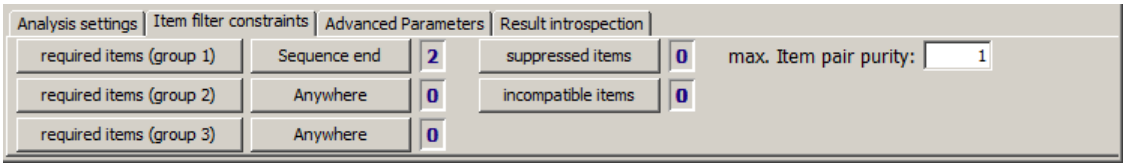
In the screenshot, the following settings were specified:

- The detected sequential patterns will be saved under the name `assoc_PURCHASES.mdl` in the current working directory. Per default, the created file will be a file in a proprietary binary format. But you could also save the file as a <TAB> separated flat text file, which can be opened in any text editor or spreadsheet processor such as MS Excel. Using the main menu item **Preferences→Sequences Preferences** you can switch the output format, for example to the intervendor XML standard for data mining models, PMML.
- The currently specified settings will automatically be saved to an XML parameter file named `assoc_params_PURCHASES.xml` every time the button **Start training** will be pressed. The resulting XML file can be reloaded in a later Synop Analyzer session via the main menu item **Analysis→Run sequences analysis**. This reproduces exactly the currently active parameter settings and data import settings.
- The patterns to be detected should consist of up to 3 parts (itemsets) involving up to 2 time steps. When specifying the parameters for a sequential patterns analysis, you should always specify an upper boundary for the desired sequence lengths, otherwise the analysis can take extremely long time.

- The patterns to be detected should contain up to 3 items. When specifying the parameters for a sequential patterns analysis, you should always specify an upper boundary for the number of items, otherwise the analysis can take extremely long time.
- The single parts (itemsets) within the patterns to be detected should consist of up to 2 items. This setting is redundant here, since we have already specified that the sequences to be detected should contains 1 or 2 time steps and the total number of items should not exceed 3. Therefore, itemsets of more than 2 items are not possible anyhow.
- The patterns to be detected should occur in at least 5 entities. When specifying the parameters for a sequences analysis, you should always specify an lower boundary for the absolute or relative support, otherwise the training can take extremely long time.
- The upper limit for the number of patterns to be detected and displayed is set to 1000. If more patterns are found, the 1000 patterns with the highest values of the measure currently specified in the selector box **Sorting criterion** will be selected. In our example, the 1000 patterns with highest lift will be selected.

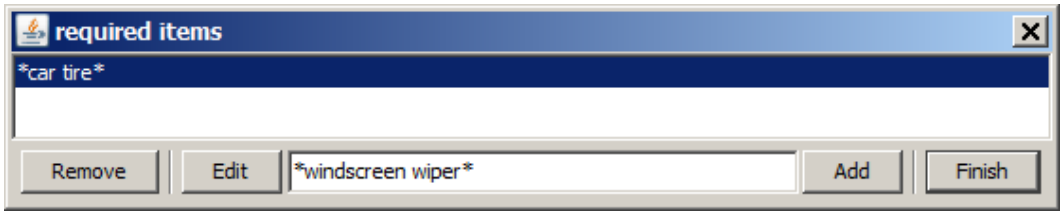
3.10.5 Pattern content constraints ('item filters')

Filter criteria defining the desired contant of the patterns to be detected can be specified using the second tab named **Item filters** of the bottom part of the sequential patterns analysis screen. The tab itself displays how many content filter criteria of the various types have been set, the specification of new content filter criteria is performed within pop-up dialogs which open up when one presses one of the buttons in the tab.



- The three buttons named **Required items (group n)** define items which must occur in each detected pattern. If several item patterns are specified within one 'required group', at least one of them must appear in each detected sequence. In the sequential patterns analysis module, up to 3 different groups of required items can be specified. The detected patterns must contain at least one item out of every specified group.
  Each item specification can contain wildcards (*) at the beginning, in the middle and/or at the end. A wildcard stands for an arbitrary number of arbitrary characters or nothing. The spelling of the items with upper case and lower case letters and empty spaces must exactly match the spelling of the field names and value names as it is displayed in the module . You can either type in the desired values into the input field, or you can select one or more values from a drop-down list of all

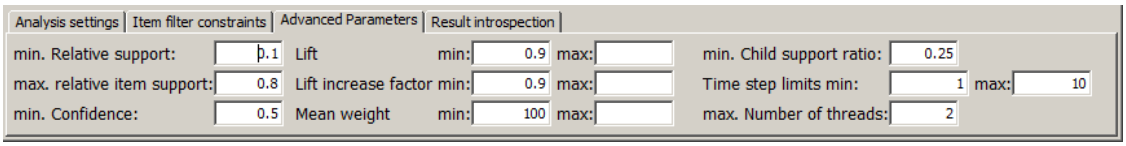
available items in the data by pressing the arrow symbol at the right edge of the input field.
As the first required item group in our example we specify '*car tire*' and '*windscreen wiper*'. That means we look for patterns which involve customers who have bought car equipment such as tires or windscreen wipers. We enter each text into the editor field of the pop-up dialog and then press **Add**. After closing the pop-up dialog we set the desired position of the required items to 'at the end of the sequence'. Hence, we want to find sequences of product purchases which lead to the purchase of car equipment at the end.



- We could specify two more groups of required items, but in our example we do not make use of this possibility.
- **Suppressed Items** are items which are to be ignored during the pattern search. In our example, we do not use this feature.
- If a pair of items or item groups has been specified as **incompatible** (by pairs), then none of the detected sequences will contain more than one item out of this set. In the text field of the pop-up dialog, you can enter several patterns, separated by comma (,) without adjacent spaces. If a pattern contains a comma as part of the pattern name, escape it by a backslash (\). Each pattern can contain one or more wildcards (*) at the beginning, in the middle and/or at the end.
  In general it is reasonable to specify items from highly correlated data fields as 'incompatible'. Otherwise one would obtain many patterns with very high lift values in which one item from each of the two highly correlated fields appears. These trivial patterns might shadow the truely interesting, non-trivial patterns. In our example, we do not use this feature.
- The **item pair purity** of two items i1 and i2, is the number of entities on which both items occur divided by the maximum of the absolute supports of the two items. Item pairs with a purity of 1 are 'perfect pairs': whenever i1 occurs on an entity, also i2 occurs in it, and vice versa.Defining an upper limit for the permitted item pair purity is therefore an alternative means for specifying many single incompatible item pairs. It serves to suppress all trivially highly correlated item pairs from the sequential patterns analysis.

3.10.6 Advanced pattern statistics constraints

The third tab at the lower end of the screen, **Advanced Parameters**, provides 9 parameters which serve for fine-tuning the detected pattern set based on certain statistical measures.



- The **relative support** of the patterns to be detected in our example must be at least 0.1, or 10% of all entities. When specifying the parameters for a sequential patterns training, you should always specify an lower boundary for the absolute or relative support, otherwise the training can take extremely long time. In our example, however, setting the minimum relative support to 0.1 has no real effect and is redundant since we have already specified a minimum absolute support of 5, which is more than 10% of all 24 entities (customers) contained in the data.

- The **relative support of an item** is the item's absolute support divided by the total number of entities. In other words, the relative support is the a-priori probability that the item occurs with a randomly selected entity value. Items which appear with (almost) every entity often represent trivial information which one does not want to find in the detected patterns. In our example, we have specified an upper boundary of 0.8 in order to suppress items which occur on at least 80% of all entities.

- The **confidence** of a sequence rule is the ratio between the rule's support and the rule body's support. An sequence rule is an sequence of n itemsets separated ny n-1 time steps in which the first n-1 of the n itemsets are considered the 'rule body' and the last itemset is considered the 'rule head'. A rule's confidence is the probability that the rule head is true if one knows for sure that the entire rule body is true. In our example we have specified that we want to search only sequences with confidence value of at least 0.2.

- Next, in our example we want to find only patterns whose lift is at least 0.9. Hence, we are interested only in 'frequent' patterns with a positive correlation of the involved itemsets in the time order defined by the sequence.

- The patterns consisting of more than two parts (itemsets) must have lift increase factors of at least 0.9. That means, a longer sequence should only be formed if the prolongation increases the positive correlation of all involved item sets.
  The specification of an upper or lower limit for the lift increase factor often is a very effective means for preventing the set of detected patterns from growing too big and for suppressing the appearance of 'redundant', trivial extensions of relevant patterns by just appending arbitrary itemsets to them.

- The **weight** of an sequence is the mean weight of all entities on which the sequence occurs. A minimum or maximum threshold for the sequences' weights can only be specified if a weight field has been defined on the input data. We specify a minimum weight of 100, that means we only want to find sequences which apply to customer groups which have a purchase history of at least 100 EUR in our supermarket.

- The parameter **minimum child support ratio** defines boundary for the acceptable 'support shrinking rate' when creating expanded sequences out of existing sequences by adding an additional item. An expanded sequence of n items will be rejected if at

least one of the possible parent sequences has a support which is so large that when multiplied with the minimum shrinking rate, the result is larger than the actual support of the expanded sequence. In our example we have specified the value of 0.25. That means we suppress the formation of patterns whose support is less than 25% of the support of the least frequent possible parent pattern.

- The two parameters named **Time step limits** permit to specify a lower and upper boundary for the duration of the single time steps which form the sequences to be detected. You should enter e pure number without a time unit. The suitable time unit is chosen automatically by the software: days if the order field contains dates, seconds if it contains time stamps, and years if it contains year numbers. In our example, we specify that the time differences between the purchases in our patterns should be between 1 and 10 days.

- The parameter **maximum number of threads** specifies an upper limit for the number of parallel threads used for reading and compressing the data. If no number or a number smaller than 1 is given here, the maximum available number of CPU cores will be used in parallel.

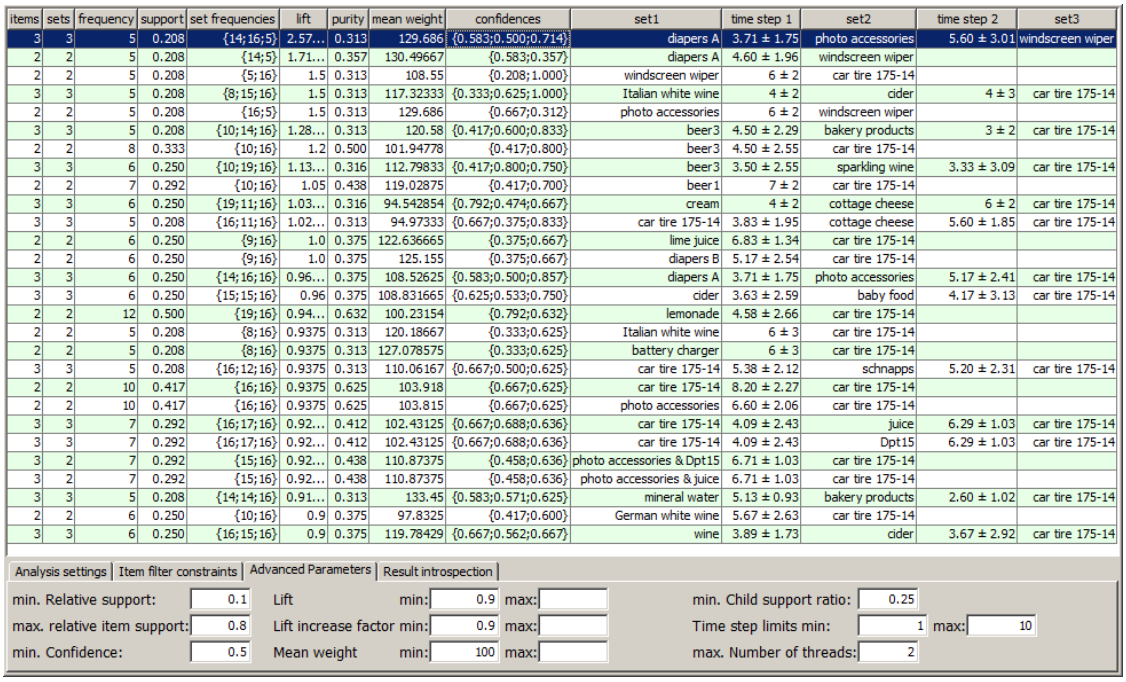
3.10.7 Result display options

The fourth tab within the tool bar at the lower border of the sequences analysis window offers some capabilites to introspect and export the generated patterns and the entities on which they appear. Some of the buttons only become enabled if you have selected one or more patterns by mouse clicks in the result table above the tool bar.

The screenshot shown below results if one performs the parameter settings described in the previous sections, presses the button **Start training** in the first tab and finally selects the first resulting pattern by left mouse click.

The tabular view of detected patterns contains the statistical measures of each pattern and its content, the itemsets which form the pattern. The most important statistical measures are, from left to right: the number of items in the pattern, the sequence length, that means the number of itemsets in the pattern, the pattern's absolute and relative support, the absolute supports of the involved itemsets, the lift, purity and weight, and finally the list of itemsets which form the pattern.

If the user has specified a time step limit in the third tab of the bottom tool bar (in our example, that has been the case), then the result table also contain time step information. Each time step information contains the mean and the standard deviation of the time measured on the training data.



items	sets	frequency	support	set frequencies	lift	purity	mean weight	confidences	set1	time step 1	set2	time step 2	set3
3	3	5	0.208	{14;16;5}	2.57...	0.313	129.686	{0.583;0.500;0.714}	diapers A	3.71 ± 1.75	photo accessories	5.60 ± 3.01	windscreen wiper
2	2	5	0.208	{14;5}	1.71...	0.357	130.49667	{0.583;0.357}	diapers A	4.60 ± 1.96	windscreen wiper		
2	2	5	0.208	{5;16}	1.5	0.313	108.55	{0.208;1.000}	windscreen wiper	6 ± 2	car tire 175-14		
3	3	5	0.208	{8;15;16}	1.5	0.313	117.32333	{0.333;0.625;1.000}	Italian white wine	4 ± 2	cider	4 ± 3	car tire 175-14
2	2	5	0.208	{16;5}	1.5	0.313	129.686	{0.667;0.312}	photo accessories	6 ± 2	windscreen wiper		
3	3	5	0.208	{10;14;16}	1.28...	0.313	120.58	{0.417;0.600;0.833}	beer3	4.50 ± 2.29	bakery products	3 ± 2	car tire 175-14
2	2	8	0.333	{10;16}	1.2	0.500	101.94778	{0.417;0.800}	beer3	4.50 ± 2.55	car tire 175-14		
3	3	6	0.250	{10;19;16}	1.13...	0.316	112.79833	{0.417;0.800;0.750}	beer3	3.50 ± 2.55	sparkling wine	3.33 ± 3.09	car tire 175-14
2	2	7	0.292	{10;16}	1.05	0.438	119.02875	{0.417;0.700}	beer1	7 ± 2	car tire 175-14		
3	3	6	0.250	{19;11;16}	1.03...	0.316	94.542854	{0.792;0.474;0.667}	cream	4 ± 2	cottage cheese	6 ± 2	car tire 175-14
3	3	5	0.208	{16;11;16}	1.02...	0.313	94.97333	{0.667;0.375;0.833}	car tire 175-14	3.83 ± 1.95	cottage cheese	5.60 ± 1.85	car tire 175-14
2	2	6	0.250	{9;16}	1.0	0.375	122.636665	{0.375;0.667}	lime juice	6.83 ± 1.34	car tire 175-14		
2	2	6	0.250	{9;16}	1.0	0.375	125.155	{0.375;0.667}	diapers B	5.17 ± 2.54	car tire 175-14		
3	3	6	0.250	{14;16;16}	0.96...	0.375	108.52625	{0.583;0.500;0.857}	diapers A	3.71 ± 1.75	photo accessories	5.17 ± 2.41	car tire 175-14
3	3	6	0.250	{15;15;16}	0.96	0.375	108.831665	{0.625;0.533;0.750}	cider	3.63 ± 2.59	baby food	4.17 ± 3.13	car tire 175-14
2	2	12	0.500	{19;16}	0.94...	0.632	100.23154	{0.792;0.632}	lemonade	4.58 ± 2.66	car tire 175-14		
2	2	5	0.208	{8;16}	0.9375	0.313	120.18667	{0.333;0.625}	Italian white wine	6 ± 3	car tire 175-14		
2	2	5	0.208	{8;16}	0.9375	0.313	127.078575	{0.333;0.625}	battery charger	6 ± 3	car tire 175-14		
3	3	5	0.208	{16;12;16}	0.9375	0.313	110.06167	{0.667;0.500;0.625}	car tire 175-14	5.38 ± 2.12	schnapps	5.20 ± 2.31	car tire 175-14
2	2	10	0.417	{16;16}	0.9375	0.625	103.918	{0.667;0.625}	car tire 175-14	8.20 ± 2.27	car tire 175-14		
2	2	10	0.417	{16;16}	0.9375	0.625	103.815	{0.667;0.625}	photo accessories	6.60 ± 2.06	car tire 175-14		
3	3	7	0.292	{16;17;16}	0.92...	0.412	102.43125	{0.667;0.688;0.636}	car tire 175-14	4.09 ± 2.43	juice	6.29 ± 1.03	car tire 175-14
3	3	7	0.292	{16;17;16}	0.92...	0.412	102.43125	{0.667;0.688;0.636}	car tire 175-14	4.09 ± 2.43	Dpt15	6.29 ± 1.03	car tire 175-14
3	2	7	0.292	{15;16}	0.92...	0.438	110.87375	{0.458;0.636}	photo accessories & Dpt15	6.71 ± 1.03	car tire 175-14		
3	2	7	0.292	{15;16}	0.92...	0.438	110.87375	{0.458;0.636}	photo accessories & juice	6.71 ± 1.03	car tire 175-14		
3	3	5	0.208	{14;14;16}	0.91...	0.313	133.45	{0.583;0.571;0.625}	mineral water	5.13 ± 0.93	bakery products	2.60 ± 1.02	car tire 175-14
2	2	6	0.250	{10;16}	0.9	0.375	97.8325	{0.417;0.600}	German white wine	5.67 ± 2.63	car tire 175-14		
3	3	6	0.250	{16;15;16}	0.9	0.375	119.78429	{0.667;0.562;0.667}	wine	3.89 ± 1.73	cider	3.67 ± 2.92	car tire 175-14

Analysis settings
Item filter constraints
Advanced Parameters
Result introspection
min. Relative support: 0.1
max. relative item support: 0.8
min. Confidence: 0.5
Lift min: 0.9 max:
Lift increase factor min: 0.9 max:
Mean weight min: 100 max:
min. Child support ratio: 0.25
Time step limits min: 1 max: 10
max. Number of threads: 2

The itemsets describing numeric data field values contain, in addition to the value range limits, an extra information within curly braces: the position of the value range within the overall value distribution of the numeric data field. For example, the text `Age=[20..30[ \=3(10)\` means that the age range from 20 (incl.) to 30 (excl.) is the third smallest out of 10 value ranges, hence the age value is below average but not strongly below average.

The numbers in the table column **set frequencies** contain the absolute supports of the different itemsets of the pattern, in the same order in which the itemset names appear in the columns at the right end of the result table.

In the tool bar tab **Result introspection** the following options are available:

- The information displayed at the left end of the tab contains the name of the data source and the number of patterns which are currently selected.
- The next vertical pair of radio buttons determines what happens if several sequences have been selectend and then the button is pressed. The button's purpose is to display those entities which support the selected sequences. The question is, does this mean the intersection or the superset of the supports of the single selected patterns? This question is answered by the choice made in these radio buttons.
- The second vertical pair of radio buttons has a similar function to the pair next to it: it specifies whether pressing the button dispays entire data sets or only the entity IDs of the entities which support the selected patterns.
- The button opens an additional window which shows the data groups on which the currently selected association patterns occur. Whether the new window contains full-width data records or only entity IDs, and whether it contains the intersection

or the superset of the data sets supporting the single selected patterns, is defined by the radio buttons described above.

- The button opens an additional window in which the data groups on which the currently selected patterns occur can be visually explored. Whether the new window contains the intersection or the superset of the data groups supporting the single selected patterns, is defined by the radio buttons described above. The new window provides the entire functionality of the module multivariate analysis.

- Using the button you can export the currently selected patterns, or all patterns if none has been selected, into a <TAB> separated flat text file, into a PMML `SequenceModel` or into a series of SQL SELECT statements.

- Using the button you can export the data groups supporting the currently selected patterns into a <TAB> separated flat text file or into a spreadsheet in `.xlsx` format.

3.10.8 Applying sequence models to new data ('Scoring')

Sequenz models can be applied to new data in order to create predictions on these data. For example, a sequence model could use the click history of a web shop user to decide which product offers or banners are to be shown to this user next. Another sequence model could serve as an early warning system in a production process, predicting upcoming problems and faulty products. This application of sequence models to new data for predictive purposes is called 'scoring'.

In the current version of Synop Analyzer, sequence models must satisfy a certain precondition for being usable for scoring: all sequences in the model must have rule heads (final parts of the sequence) containing values of one single data field. This data field is called the target field of the model. In the sample applications cited above (web shop, production monitoring), the target fields could be `ARTICLE` or `ERROR`.

If all rules of the model only contain information ('items') from one single data field, the precondition for scoring is trivially satisfied. If not, you can enforce the precondition by defining one or more required items of type **Sequence end** when training the model. In this case, you must make sure all required head items are values or value ranges of one single data field.

You load and apply a sequence model by first opening and reading the new data, by then pressing the button

in order to start the sequential patterns analysis module and by then clicking the button **Load model** in the tab **Scoring Settings** of the tool bar at the lower end of the panel's GUI window.

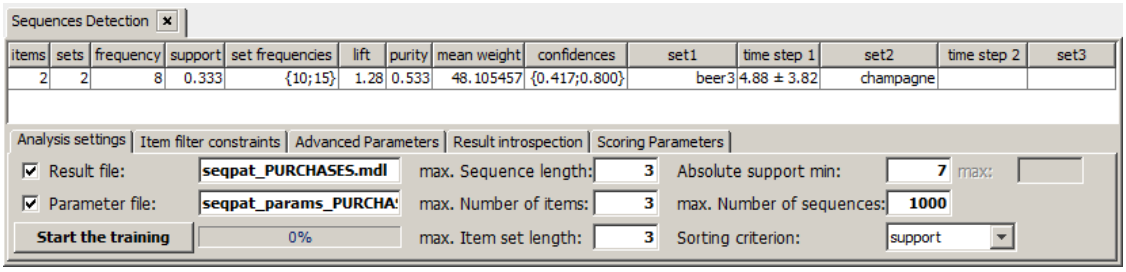
In the following sections we will demonstrate the process of sequence rule scoring with the help of a concrete example use case: using a sequence model we want to identify suitable customers for a marketing campaign for a certain premium product: champagne.

For this purpose, we load the sample data `doc/sample_data/RETAIL_PURCHASES.txt`. We assume that these data have been imported into Synop Analyzer as described in Name mappings and Taxonomies, that means with `PURCHASE_ID` as group field, `CUSTOMER_ID` as entity field, `DATE` as order field, `PRICE` as weight field and with `doc/sample_data/RETAIL_NAMES_DE_EN.txt` as article names and `doc/sample_data/RETAIL_ARTICLEGROUPS.txt` as article hierarchies.

Then we start the sequential patterns analysis module. We first want to train a sequence model and then apply it. We specify the following settings for the model to be created:

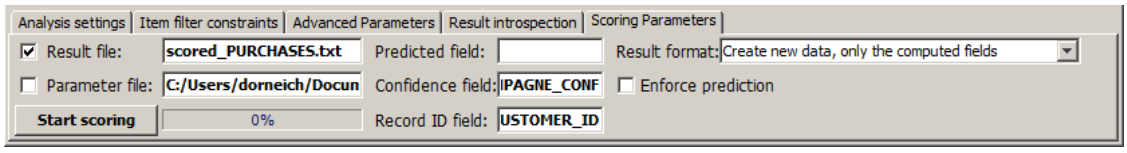
- Required item `champagne` of type **Sequence end**,
- In the toolbar tabs **Analysis settings** and **Advanced Parameters**, we specify a minimum absolute support of 7, a minimum lift of 1.2, minimum lift increase factor of 1.0 and a permitted time step size between 1 and 14 days.

The sequence model trained with these settings contains one single sequence. The sequence states that customers who have purchased a specific beer ('beer 3') have a probability of 80% for purchasing champagne in the 1 to 14 days after buying the beer.



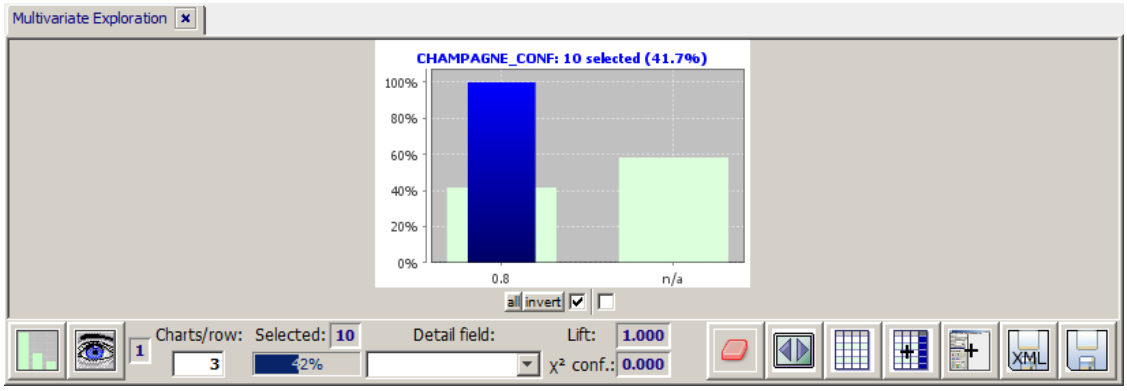
Now we want to use the generated model for identifying the most susceptible customers for an advertizing campaign for champagne within our small sample database `RETAIL_PURCHASES.txt` of 24 customers.

We move to the tab **Scoring Parameters** in the tool bar of the sequential patterns analysis module. Here, we enter the name of the file in which the scoring results are to be stored (`scored_PURCHASES.txt`), we define the scoring result data fields to be contained in that file and we specify that the new file should be a copy of the existing file in-memory data source plus the new computed data fields. (**Create new data, original plus computed fields**).Since all sequence rules in our model predict the same value (`champagne`), we do not need a new data field **Predicted field**. Instead, we are interested in the predicted probability of that value, therefore we define a **Confidence field** and call it `CHAMPAGNE_CONF`. For being able to identify the single customers in the new data, we make sure the group field `PURCHASE_ID` (and automatically also the attached entity field `CUSTOMER_ID`) is contained in the new data.

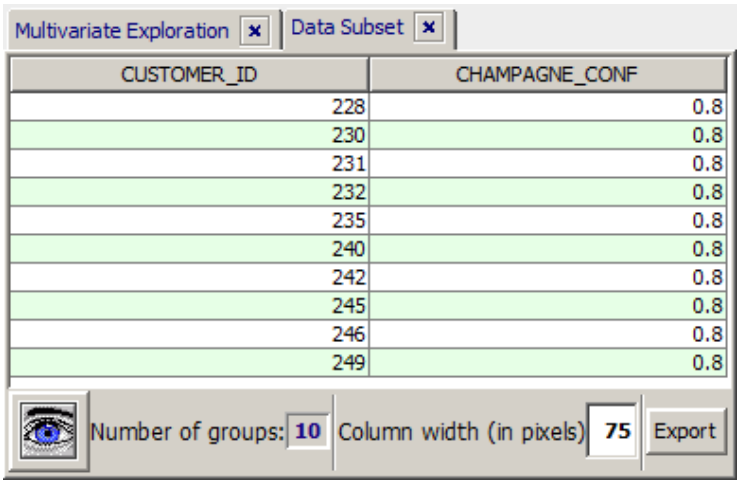


By means of the button **Start scoring** we create the scoring results, write the desired result file to disk and open the resulting data as a new in-memory data source in Synop Analyzer, that means as a new tab in the left column of the Synop Analyzer workbench.

We introspect the scoring result data with the module 'multivariate exploration'. We see that the model has identified 10 of the 24 customers as sucsceptible for champagne:



Via the button we submit the selected 10 customer IDs to a last visual examination. Then we can use the button **Export** to save the resulting list to a flat file or Excel spreadsheet, or we can use the main menu button **Report** to create a HTML or PDF report.



3.11 The Self-Organizing Maps (SOM) module

3.11.1 Purpose and short description

Self-organizing maps (SOM) are neural networks in which the neurons form a two-dimensional square grid or a hexagonal grid and each neuron is connected by artificial synapses to its near neighbors. A SOM is trained in an unsupervised learning process on a so-called training data set. Each neuron has a set of properties - the so-called weights - which corresponds to the set of data attributes available in the training data, and each neuron represents a unique combination of values of these attributes.

The purpose of the SOM is to define a mapping from the high-dimensional training data space with its many attribute dimensions to a two-dimensional representation which is easy to visualize and interpret but which conserves as much as possible of the structural (topological) information of the original data space.

There are two major application areas for SOM models: data visualization and data clustering on the one hand and scoring (prediction of unknown attribute values) on the other hand. In this latter case, the trained SOM model is applied to a neu data collection, the so-called scoring data, in which some of the attributes or attribute values of the original training data are missing.

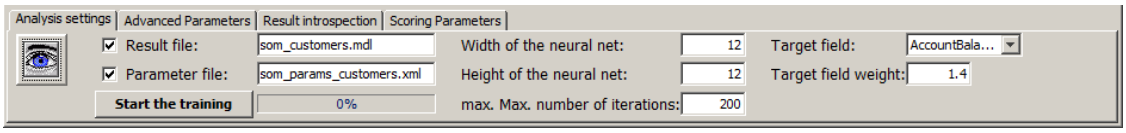
You can find more details on the theoretical approach and links for further reading on http://en.wikipedia.org/wiki/Self-organizing_map.

3.11.2 Basic parameters for SOM trainings

In Synop Analyzer, a SOM training is started by loading a data source - the so-called training data - into memory and by clicking on the button in the input data panel on the left side of the Synop Analyzer GUI. The button opens a panel named **SOM Training**. In the lower part of this panel, you can specify some parameters for the next SOM training and start the training process. The training process itself can be a long-running task, therefore it is executed asynchronically in one or more parallelized background threads. After the end of the training, the resulting SOM model will be displayed in the upper part of the panel.

The following paragraphs and screenshots demonstrate the handling of the various subpanels and buttons at hand of the sample data `doc/sample_data/customers.txt`. We assume that these data have been read into memory without changing any default settings in the data import panel on the left side of the screen.

The first visible tab in the toolbar at the lower end of the SOM panel contains the most important parameters for SOM trainings.



In the screenshot, the following settings were specified:

- The button serves to restrict the set of data fields which will be used for the model training. In our example, we do not use this feature.
- The trained SOM model will be saved under the name `som_customers.mdl` in the current working directory. Per default, the created file will be a flat file in a proprietary binary data format, which can only be opened and reused in Synop Analyzer. Using the main menu item **Preferences→SOM Preferences** you can switch the output format to the intervendor XML standard for data mining models, PMML.
- The currently specified settings will automatically be saved to an XML parameter file named `som_params_customers.xml` every time the button **Start training** will be pressed. The resulting XML file can be reloaded in a later Synop Analyzer session via the main menu item **Analysis→Run SOM training**. This reproduces exactly the currently active parameter settings and data import settings.
- The size of the neural net is set to 12 * 12 neurons, which are placed into a square grid.
- The number of training iterations during the SOM training process is limited to 200. In each iteration, the neural net learns each data record once, each data record is assigned to the neuron which best represents the properties of the data record, and then the weights (properties) of the best matching neuron itself and its nearest neighbors are shifted towards the properties of the assigned data record. This is the way the SOM net 'learns the data'. The training ends when either no further optimization of the mapping quality of the net can be reached or if the maximum number of iterations has been reached.
- When training a SOM model, one can optionally specify a **target field**. That means one informs the model on the fact that the model will later be used to predict this data field on new data.
- Using the parameter **Target field weight** you can overweight the target field relative to the other data fields when training the SOM by specifying a weight factor larger than 1. As a consequence, the resulting SOM model will fit the values of the target field particularly well, at the cost of some loss of fitting quality on the other data fields.

You should consider specifying a target weight larger than 1 if you want to train a SOM for predicting a target field and if with the default training settings the resulting SOM target field shows no clear structures but rather an amorphous green and grey pattern.

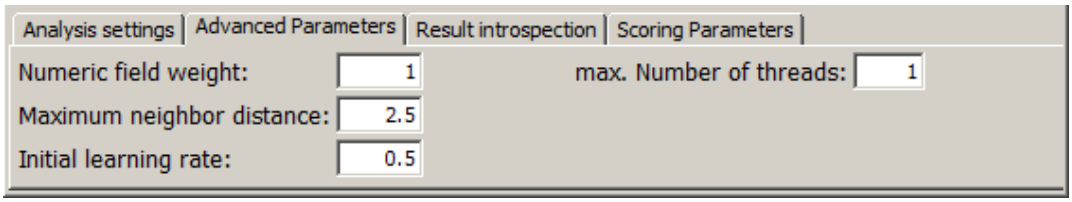
On the other hand, one can easily generate an 'over-trained' model by pushing the target weight to high. Over-training means that the resulting SOM almost perfectly maps all

record's target field values but performs poorly both on the other data fields on the training data and when predicting the target field values of new scoring data.

It is always a good idea to put aside a small part of the available training before starting the SOM training. These data can then be used to validate the SOM. That means one lets the model predict the target field values and compares the predictions to the actual target field values. This approach helps to find the training parameter settings which produce the model with the smallest mean squared difference between the actual and the predicted target field values.

3.11.3 Expert parameters for SOM trainings

The second tab at the lower end of the screen, **Advanced Parameters**, provides 4 parameters which serve for fine-tuning the training process. You should only modify them if you are familiar with the SOM approach and algorithm parameters such as 'learning rate' or 'neighborhood radius'.



- **Numeric field weight**: Per default, each numeric data field contributes with the same weight factor (of 1) to the distance calculations between neurons and data records as the Boolean and textual fields. You can define a higher or lower weight factor for the numeric fields compared to Boolean and textual fields using this parameter. Note that weight settings for specific fields, for example the target field weight, overwrite this general setting, the weight factors are not multiplied.

- The **maximum neighbor distance** is the Euclidean distance ($dx^2 + dy^2$) between neurons up to which learned information is distributed from the best matching neuron to neighbored neurons of that neuron. If that value is 1.5, for example, then 8 neighbored neurons are influenced by each assignment of a data record to its best matching neuron, namely the neuron's 4 nearest neighbors at distance 1 and 4 second nearest neighbors at distance 1.41. During the SOM training process, the maximum neighbor distance is reduced step by step.

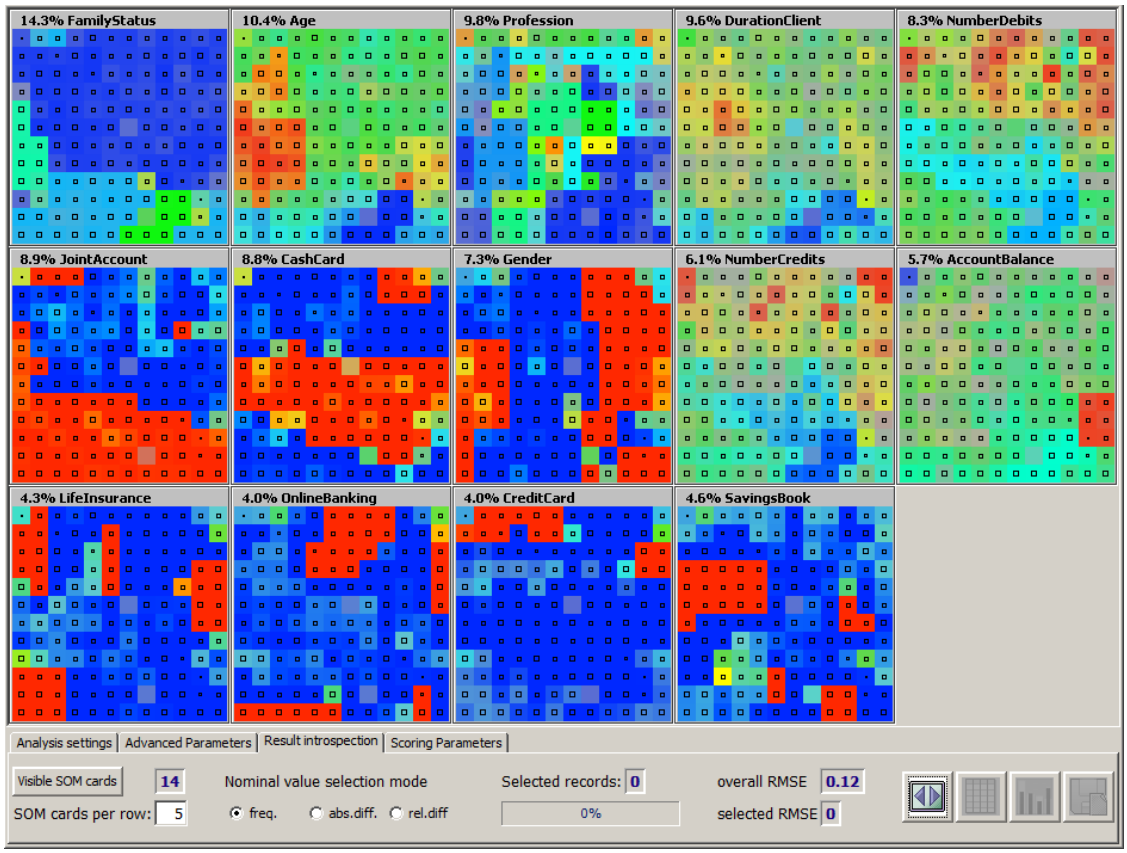
- The **initial learning rate** is the strength of change of a neuron's properties when a new data record is being learned. If the learning rate is 0.5, for example, and if the best matching neuron of a data record with `Age=46` has the property `Age=40` before learning the record, the neuron's property will have changed to `Age=43` after learning the data record.

- The parameter **max. number of threads** specifies an upper limit for the number of parallel threads started by the SOM training engine in order to perform the training. If this input fields contains a value of 0 or smaller, the software is free to fully exploit the available CPU, that means to start one thread on each CPU core of the computer.

3.11.4 Interpreting the result visualizations

The third tab within the tool bar at the lower border of the SOM training window offers some capabilites to modify the display mode of the created SOM model and to introspect and export the model itself or certain data clusters marked on it. Some of the buttons only become enabled if you have selected one or more neurons by mouse clicks within the SOM cards.

The screenshot shown below results if one performs the parameter settings described in the previous sections and then presses the button **Start training**.



The main part of the screen displays one separate map, a so-called SOM card, for each data field. The SOM cards can be interpreted as follows:

- The title row of each card shows the name of the data field which the card describes as well as an importance percentage number which indicates how important the data field is for the SOM model. The sum of all importance numbers is always 100%. A high value indicates that the SOM model is able to predict the values of this data field on almost all training data records with high accuracy and confidence and that the SOM card shows a clear structure of large and homogeneous regions. Small importance numbers result from SOM cards which look like rag rugs of which have many grey spots, which indicates that the data records mapped to these neurons have a rather diffuse value range,

- Each single small uniformly colored squaree within a SOM card represents one neuron. Hence, all SOM cards in our example consist of 12 * 12 small colored squares.

- The black dots or black quadrangles in the center of each colored square indicate how many training data records have been mapped to this neuron. A small dot represents one single data record or a very small number of data records, the longer each side of the quadrangle is, the more data records have been mapped to the neuron. You can hide this additional information by right-clicking on one of the SOM cards while keeping the <Ctrl> key pressed.

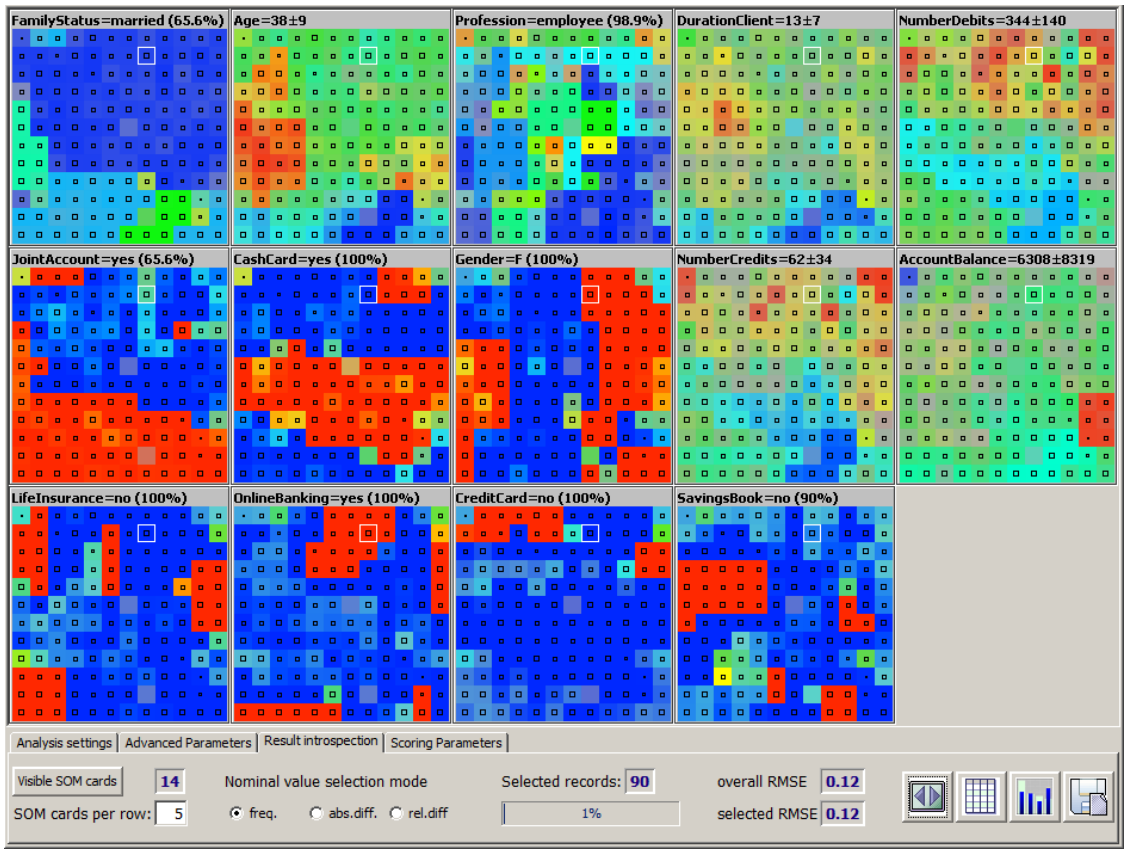
- The color coding scheme of the SOM cards representing numeric data fields correspond to the familiar color coding of topographical maps: low values are blue, medium values green and high values red.

- In the SOM cards for textual data fields, for example in the card for the field `FamilyStatus` in our example, the most frequent value (`married`) is dark blue, the second most frequent value light blue, the third one turquoise, and so on through the rainbow. The least frequent field values are orange or red.

- In the SOM cards for Boolean, that means two-valued fields such as the field `Gender`, the majority value is blue, the minority value red.

- The intensity of each colored square indicates how precisely and reliably the data field values of the training data records mapped to the neuron represented by the square coincide with the neuron's own value for the data field. The more precise and reliable the mapping, the higher the intensity. If the square is gray then the data records mapped to the neuron have a standard deviation or value distribution which is as diffuse as on the entire training data.

Synop Analyzer's SOM cards provide a wide variety of mouse-based interactivity and selection features:

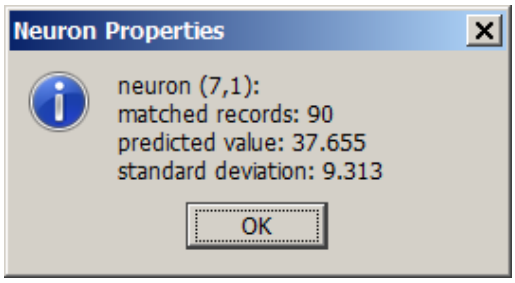


- By left-clicking one of the colored squares in one of the SOM cards, the corresponding neuron is selected on all visible SOM cards. The title information of each SOM card changes and shows the statistical properties of the training data records which have been mapped to the selected neuron. Additionally, the bottom tool bar shows the absolute and relative number of data records mapped to the selected neuron. In the picture shown above, a neuron in the middle of the second row from the top of the SOM card has been selected.

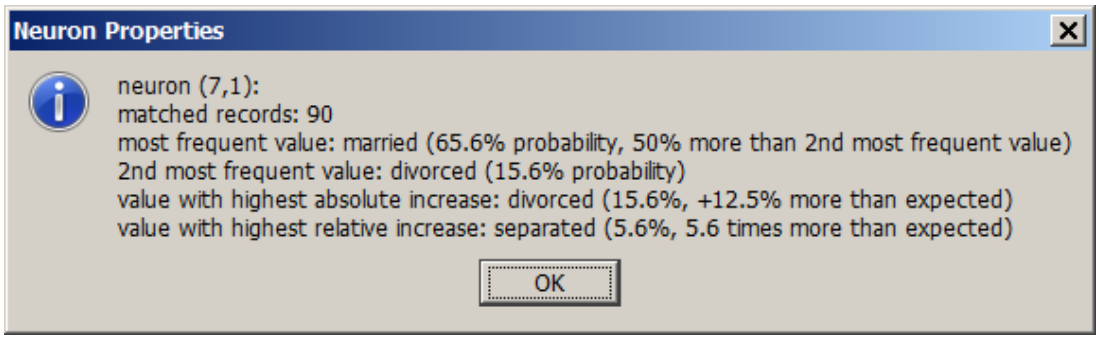
- Left-clicking a colored square while keeping the <Ctrl> key pressed adds a new selected neuron to the current selection. That means, you can select more than one neuron at once.

- Left-clicking a colored square while keeping the <Shift> key pressed selected large regions of neurons at once. More precisely, the click starts a 'flooding' algorithm which selects all neurons starting from the current position in every direction until the end of the SOM card is reached or an already selected neuron is reached.

- Right-clicking a colored square within a SOM card opens a pop-up dialog in which the statistical properties of the neuron and the training data mapped to it are shown in detail.
  For numeric data fields, the pop-up window shows the mean and standard deviation of the data field values of all training data records mapped to the neuron.



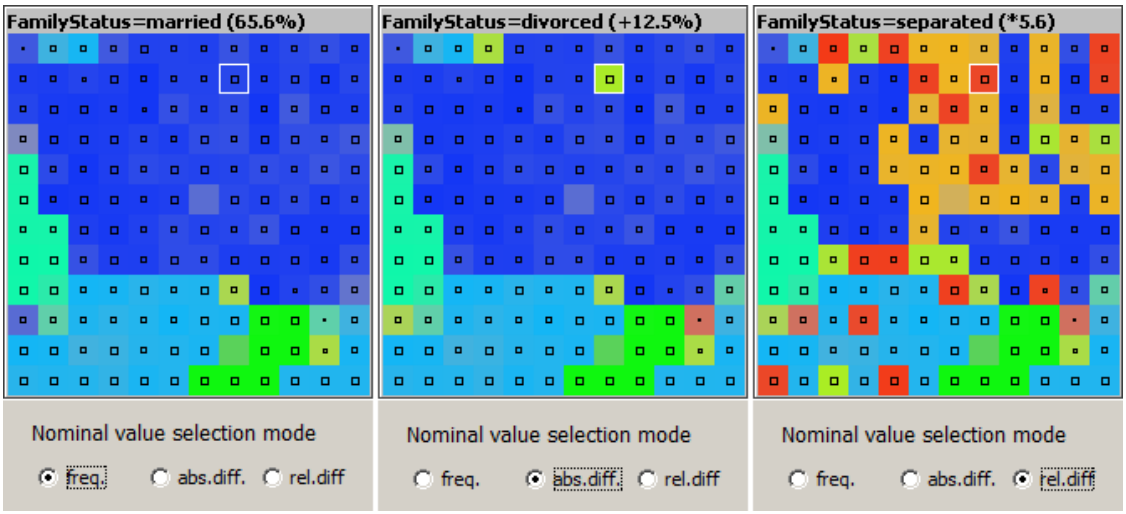
For textual data fields, the pop-up window shows the most frequent value with its percentage of occurrence, and the values which have the greatest increase rate on the neuron compared to the entire training data. There are two increase rates: an absolute or additive one (the added percentage rate), and a relative or multiplicative one (the multiplication factor of occurrence probability):


Neuron Properties
neuron (7,1):
matched records: 90
most frequent value: married (65.6% probability, 50% more than 2nd most frequent value)
2nd most frequent value: divorced (15.6% probability)
value with highest absolute increase: divorced (15.6%, +12.5% more than expected)
value with highest relative increase: separated (5.6%, 5.6 times more than expected)
OK

- Right-clicking a square while keeping the <Ctrl> key pressed switches the additional occupation frequency information, that means the black dots and quadrangles, on or off

In the tool bar tab **Result introspection**, the following options are available:

- The button **Visible SOM cards** opens a pop-up dialog in which you can restrict the set of data fields whose SOM cards are to be shown on screen. The blue number at the right side of the button displays the number of currently visible SOM cards.
- The input field **SOM cards per row** specifies how many SOM cards will be shown within one screen row. Hence, the field defines how big each single SOM card will appear on screen.
- The three radio button labeled **Nominal value selection mode** permit to switch between three display modes within the SOM cards for textual data fields. The default mode is shown on the left side of the picture below. In this mode, each square (neuron) is colored according to the most frequently adopted field value on all training data records mapped to this neuron, independent from the fact whether or not this value's occurrence rate is greater or smaller than the value's occurrence rate on the entire training data. In the neuron selected in the picture below, this majority value is the value `married` with an occurrence rate of 65.6%.



In the display mode **Absolute difference**, the neuron is colored like the value which has the highest absolute (additive) increase rate on the neuron compared to the entire data. That can but need not be the majority value. In our example, it is the value `divorced`, which has an occurrence rate of 15.6% on the neuron, hence an increase of 12.5 percent points compared to its overall occurrence rate of 3.1%.
In the third display mode **Relative difference**, the neuron is colored like the value which has the highest relative (multiplicative) increase factor on the neuron compared to the entire training data. In our example, it is the value `separated`. This value occurs 5.6 times more probably on the data records mapped to the neuron than on the entire data, where it occurs in only 1% of all data records.
From the picture shown above it becomes visible that the first display mode favors the most frequent field values, in the second mode also moderately frequent values have a chance to appear, and in the third mode the least frequent values are favorized because it is easier to reach high multiplication factors of occurrence when starting from a small base.

- The output field **Selected records** and the percentage bar displayed below the field show the absolute and relative size of the data subset which has been mapped to the currently selected neurons.

- The output field **Overall RMSE** contains a measure for the average accuracy of the mapping induced by the SOM, that means the mapping from the n-dimensional input data space to the two-dimensional neural network. RMSE stands for root mean squared error, that is the square root of the average over the squared mapping errors, where the squared mapping error between a neuron an a data record is the average over all squared differences between the neuron's value for each data field and the field's value on the data record. RMSE is scaled such that a value of 1 corresponds to a useless or trivial SOM model in which all neurons have identical properties: the adopt the field's mean value for numeric fields and they adopt a value occurrence distribution equal to the overall distribution on the training data for the non-numeric fields. A value of 0 stands for a perfect SOM which has no mapping error at all.

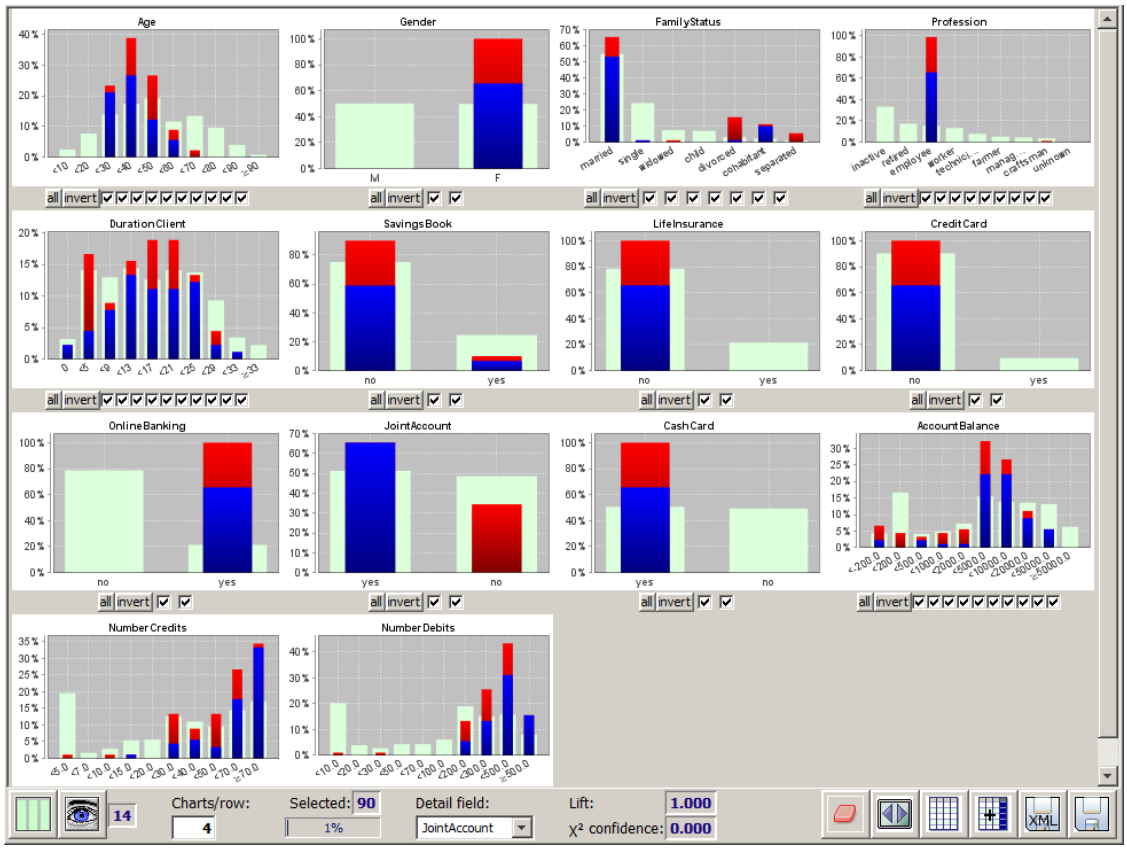
- The output field **Selection RMSE** contains the corresponding measure to overall RMSE for the subset of the training data which has been mapped to the currently selected neurons.

- By clicking on this button you re-draw all SOM cards, thereby adapting their size to the current screen width.

- opens a new panel which contains the data records which have been mapped to the currently selected neurons in tabular form. In the panel, you can sort the selected data by any data field and export the extire selection or a subset into a flat file or spreadsheet.

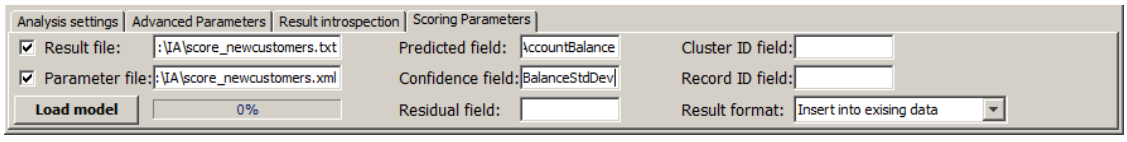
- opens an additional window in which the data groups mapped to the currently selected neurons can be visually explored. (See picture below.)
  The new window provides the entire functionality of the module multivariate analysis. The screenshot shown below explores the 90 data groups which have been mapped to the neuron which has been taken as our example selection in the previous pictures. Additionally, we have chosen the data field `JointAccount` as detail structure field. Now the blue and red bars are indicating how the dagree of usage of a joint account coincides with age, family status, account balance etc. on the selected customers.

- Using the button you can export the data groups mapped to the currently selected neurons into a <TAB> separated flat text file or into a spreadsheet in `.xlsx` format.

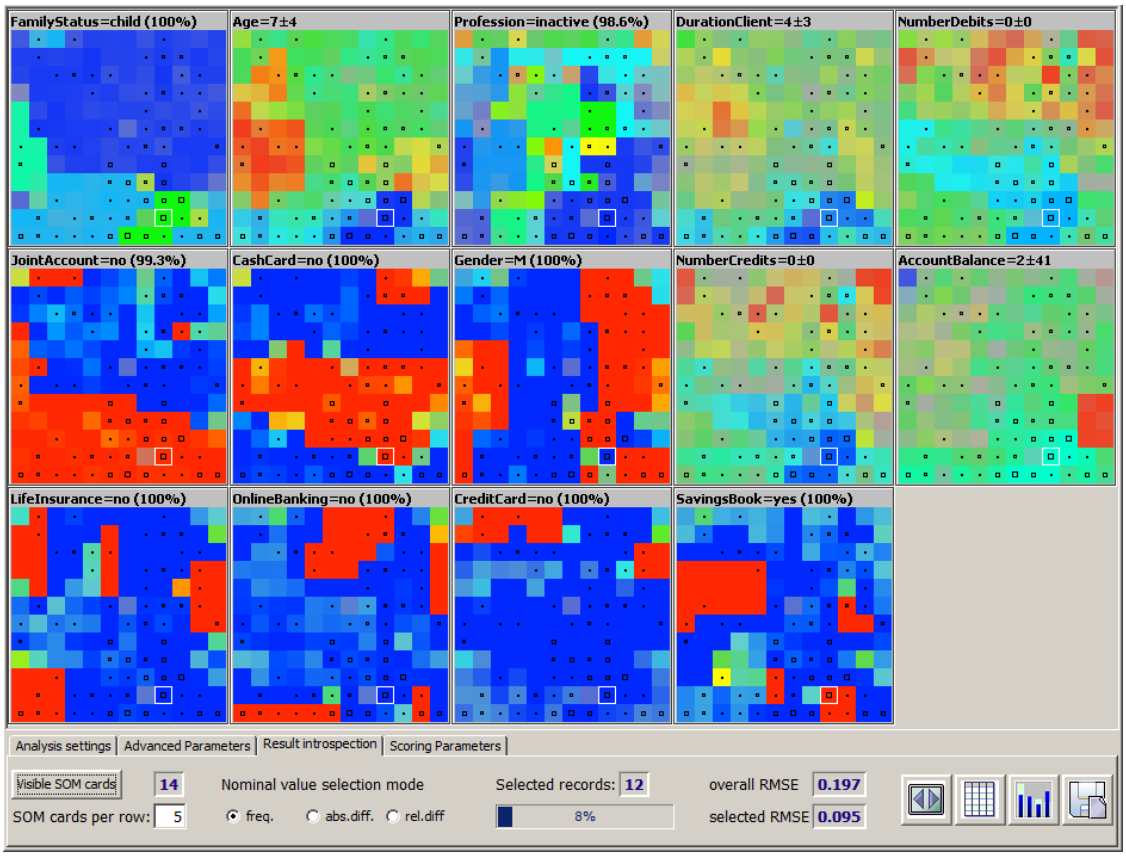


3.11.5 Apply SOM models to new data

SOM models which have been trained and stored earlier can later be reloaded and applied to a new data source. Synop Analyzer then compares the data fields available in the new data and the data fields used in the SOM model. Applying the model to the new data is only possible if at least half of the data fields used in the model are available in the new data. You load and apply a SOM model by first opening and reading the new data, by then pressing the button in order to start the SOM module and by then clicking the button **Load model** in the fourth tab of the tool bar at the lower end of the SOM panel's GUI window.



Once the SOM model has been loaded and applied successfully, the same SOM cards appear that you have seen at the end of the training process on the training data. But the black dots and quadrangles within the cards now represent the mapping of the new data records to the neural net. Correspondingly, the mapping quality measures **Overall RMSE** and **Selection RMSE** as well as the displayed relative and absolute numbers of selected data records shown in the panel's bottom tool bar now refer to the new data.



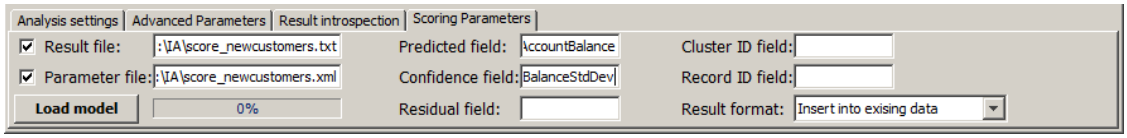
When applying a SOM model to a new data source, you should always have a look at the measure **Overall RMSE**. If this value is much larger on the new data than it was on the training data, the new data do not match the model very well, indicating that between the training data and the application data, some major shift in the rules and relations which interrelate the different data fields and their values has occurred. Hence, using this SOM model for scoring the new data, that means for predicting missing field values, can yield misleading results.

In our example, we see from the distribution of the black quadrangles that the average demographic properties of the new customers do not coincide with the average demographic proberties of the existing customer base - new customers are mostly children or young adults. But nontheless, the model seems well applicable to the new data because the overall RMSE value is only slightly larger than it was on the training data, and it is still close to 0.

3.11.6 Creating scoring results

Now we want to use the loaded SOM model for scoring, more precisely for predicting the average account balance that we can expect from each new customer after a few months of getting into business with him or her. This information can be important for customer relationship management aspects and for optimizing the bank's internal

refinancing strategy. The tab **Scoring Settings** within the SOM panel's bottom tool bar offers the following customization parameters for the SOM scoring:

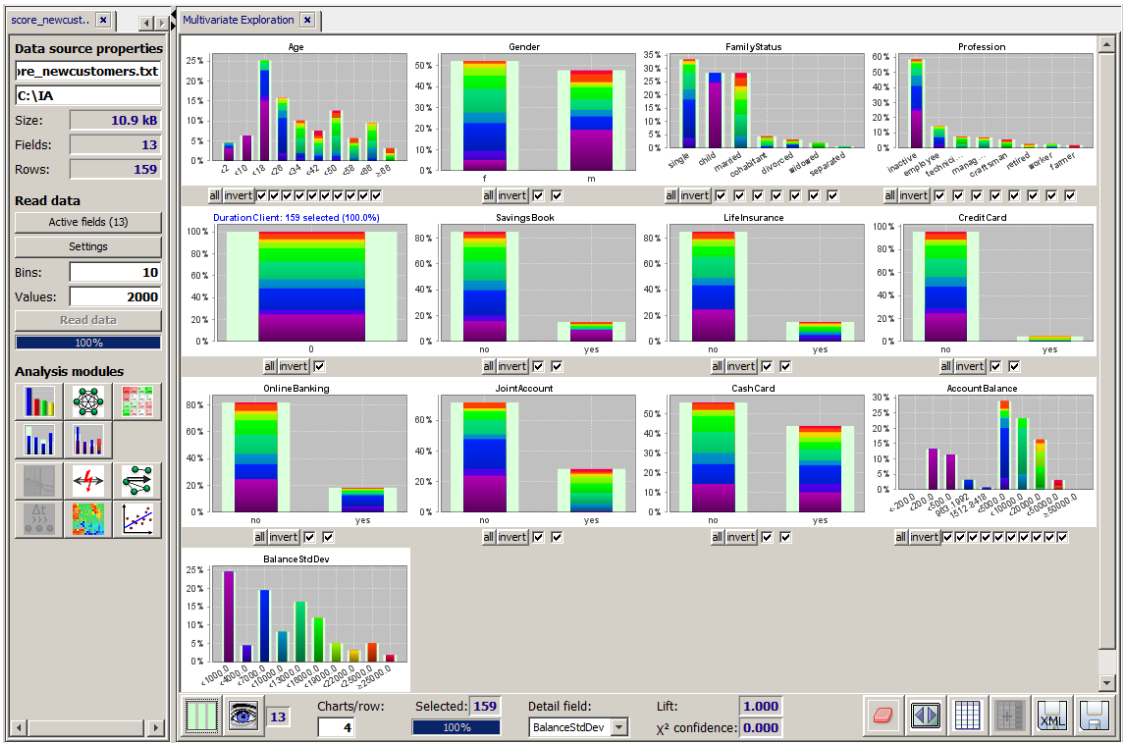


- Using the input field **Result file** you can enter the name of a flat file into which the scoring result data are to be written.
- Analogously, you can specify the name of an XML file which will persistently store the current SOM settings using the input field **Parameter file**.
- The button which previously showed the label **Load model** now displays the text **Start scoring**. By pressing this button, you start the scoring process after entering all desired customization settings.
- The next five input fields serve to define the names of computed data fields which will be added to the data and which will contain different scoring results. Normally, you are interested in only two or three of the available scoring results, then you should leave the other field names empty. The five different possible scoring result fields are the following:
  - **Predicted field** is the name of the data field into which the predicted values of the field will be written which has been specified as the target field when the SOM model was trained. In our case, the model's target field was `Kontensaldo`; in the new data, a field with this name is missing, therefore we choose exactly this name for the predicted field.
  - **Confidence field** is the name of the data field into which the model's self-estimation of the accuracy of each record's target field prediction will be written. If the target field is numeric, the confidence field will contain the estimated mean prediction error (standard deviation). For textual target fields, the confidence field will contain the estimated probability that the predicted value is the correct one. In our example, we are interested in this information and call the field `BalanceStdDev`.
  - **Residual field** is the name of the data field into which the difference between the predicted and the actual target field value will be written. Activating this field only makes sense on validation data which already contain target field values before the scoring and on which the scoring is started for model validation purposes. Therefore, we leave the field name empty in our example.
  - **SegmentID field** is the name of the data field into which for each data record the number of the best matching neuron will be written. This field is only of interest of the SOM scoring is started with the aim of clustering the data. This is not the case in our example, therefore we leave the field empty.

  - **RecordID field** is the name of the data field into which the SOM scoring engine will write the group field value of each scored data record, or, if no group field has been defined, a record ID running from 1 to the number of records in the application data. This field is important if the scoring results are to be written into a completely new data file and not to be merged into the existing data. In the first case, one normally needs a sort of primary key in the newly created data file for later being able to combine and join the new data with existing data sources. In our example write the scoring results directly into the existing data, therefore we do not need this field.

- The last selection field in the tab, **Result format**, specifies whether the newly created scoring results are to be merged into the existing data or written into a completely new data file, and if the latter is the case, whether the new file shall only contain the newly computed scoring result fields or also the preexisting data fields of the application data.

Once all settings and customizations have been performed, pressing the button **Start scoring** executes the scoring process. When the process has terminated without an error, the scoring result data are automatically opened in a new input data tab within the left screen column of the Synop Analyzer workbench. You can now apply all available analysis modules provided by your Synop Analyzer license to these new data. In the screnshot shown below, we have opened the scoring result file of our example in the module **Multivariate Exploration**.



We have selected the new data field `BalanceStdDev` as the detail structure field of our visualization. This field contains the SOM model's self-estimation on the accuracy of each

of its predictions. Blue or violett values correspond to low incertitude ranges, orange and red values to very high incertitude ranges. For some data records, the SOM thought that its prediction was very accurate - up to some 100 EUR - for other data records the model gave an incertitude range of up to 30,000 EUR.

In our example we understand that the average balance of children can be predicted quite precisely and that surprisingly the self-estimated prediction quality for men is more often very low or very high than for women, where medium incertitude ranges dominate.

By pressing the button one can introspect the entire scoring results in tabular form, sort and filter them and export parts or all of them into different persistent target formats such as flat text files or spreadsheets. In the picture shown below we have sorted the scoring results by decreasing predicted value. We see that the model predicts the highest account balances for 40 to 55 years old engineers, freelancers, craftsmen and farmers and for pensioners.

Age	Gender	FamilyStatus	Profession	Duration...	SavingsB...	LifeInsur...	CreditCard	OnlineBan...	JointAccount	CashCard	AccountBal...	BalanceStdDev
47.0	m	married	farmer	0.0	no	yes	no	yes	yes	no	32122.836	29280.752
40.0	m	married	farmer	0.0	no	no	no	no	yes	yes	27694.662	32386.254
38.0	m	married	craftsman	0.0	no	no	no	no	yes	yes	27694.662	32386.254
57.0	m	single	farmer	0.0	yes	yes	no	no	no	yes	23877.875	24369.416
67.0	f	married	retired	0.0	no	no	no	no	yes	yes	22866.98	24492.71
28.0	m	cohabitant	inactive	0.0	no	no	no	no	yes	yes	19662.33	22528.986
76.0	f	married	inactive	0.0	yes	no	no	no	no	no	18692.092	24010.508
66.0	m	married	retired	0.0	no	no	no	no	yes	no	17913.648	19347.943
63.0	f	married	inactive	0.0	no	no	yes	no	yes	yes	17826.914	21642.994
56.0	m	divorced	retired	0.0	no	no	yes	no	no	yes	17662.932	18640.105
62.0	f	married	retired	0.0	no	no	no	no	yes	no	16940.25	18423.541
39.0	f	married	employee	0.0	no	yes	yes	no	yes	yes	16281.233	21401.617
65.0	f	married	inactive	0.0	no	yes	no	no	yes	no	15372.369	14660.232
33.0	f	married	inactive	0.0	no	yes	no	no	yes	no	15372.369	14660.232
24.0	f	single	technician...	0.0	no	yes	no	no	yes	yes	14030.712	13499.906
38.0	f	married	manager,f...	0.0	no	no	no	no	yes	yes	13722.63	16301.43
61.0	f	married	inactive	0.0	no	no	no	no	yes	yes	13722.63	16301.43
26.0	f	cohabitant	employee	0.0	yes	no	no	no	yes	no	13081.785	18170.0
47.0	f	cohabitant	employee	0.0	yes	no	no	no	yes	no	13081.785	18170.0
29.0	f	single	technician...	0.0	no	yes	yes	no	no	yes	12514.346	19268.887
45.0	m	married	craftsman	0.0	no	yes	yes	no	yes	yes	12269.638	16209.663
79.0	f	widowed	inactive	0.0	yes	no	no	no	no	no	12234.491	13137.022
64.0	f	widowed	craftsman	0.0	no	no	no	no	no	no	11698.3545	15803.86
65.0	f	widowed	inactive	0.0	no	no	no	no	no	no	11698.3545	15803.86
26.0	f	married	technician...	0.0	no	yes	no	no	yes	no	11415.589	15627.377
16.0	m	single	manager,f...	0.0	no	no	no	no	no	yes	11151.924	14583.49
40.0	m	single	technician...	0.0	no	no	no	no	no	yes	11151.924	14583.49
43.0	m	married	technician...	0.0	no	no	no	no	yes	yes	10993.729	12897.743
23.0	f	married	worker	0.0	no	no	no	yes	yes	no	10679.87	17710.666
32.0	f	cohabitant	technician...	0.0	no	no	no	no	yes	no	10353.911	15142.255

Number of records: 159 Column width (in pixels) 75 Export

3.12 The Regression Analysis panel

3.12.1 Purpose and short description

A regression analysis finds a formula which predicts the value of one single data field, the so-called **target field**, as a function of other data fields, the so-called **predictor fields**. The formula is detected during a so-called training process on data, on which both the target field and the predictor fields are filled with values. The resulting formula is also called a **regression model**. The regression model can later be applied to new data in which the target field values are missing in order to predict the target field values. This step is called **scoring**.

Synop Analyzer provides several methods for creating more general regression models, for example the neural SOM method. This chapter, however, shall be focussed on **linear regression** and **logistic regression**.

A linear regression model is a linear formula which predicts the target field value y of a numeric target field from n predictor field values x_1 bis x_n:

$y = c_0 + c_1\ x_1 + ... + c_n\ x_n$.

In logistic regression, the probability of one of the two values of a two-valued target field t (the so-called '1'-value) is expressed as a formula of the kind:

$\text{proba}(t=1) = 1/(1+e^{b0+b1*x1+...+bn*xn})$.

Does every predictor field contribute exactly one regressor x_i? This is only the case for numeric and Boolean data fields. More precisely, the following holds:

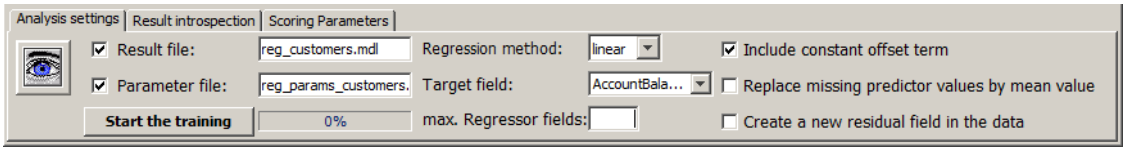
- Each numeric predictor field contributes exactly one regressor x_i.
- Each non-numeric predictor field with $n > 2$ different values contributes n regressors, one for each field value. If the regression formula is applied to a concrete data record in order to predict the target field value, only one of these n regressors contributes its coefficient c_i to the calculated result, namely the one for the field value which actually occurs in the data record.
- For Boolean fields, we assume that the regressor for the more frequent of the two values is zero. This can always be achieved by adding its real value to the constant offset c_0. The only remaining regressor for the field then captures the difference in the predicted value which results if the Boolean field does not assume its majority value but the less frequent value.

Training a linear or logistic regression model means detecting the best coefficient values c_i such that the resulting formula minimizes the mean squared difference between the actual and the predicted target field values on the training data.

Within Synop Analyzer, a linear or logistic regression analysis is started by pressing the button in the left screen column.

3.12.2 Parameters for regression analsis

The first visible tab in the toolbar at the lower end of Synop Analyzer's linear regression panel contains the available parameters for linear regression analysis.



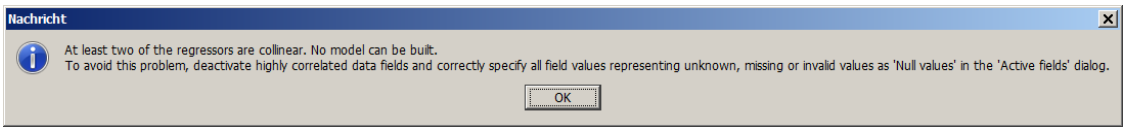
In the following, we want to explain the process of training and interpreting a regression model at the hand of a concrete example operating on the sample `doc/sample_data/-customers.txt` using the following settings:

- The button serves to restrict the set of data fields which will be used for the model training. In our example, we do not use this feature and work with all data fields.
- The resulting regression model will be saved under the name `reg_customers.mdl` in the current working directory. Per default, the created file will be a file in a proprietary binary format. But you could also save the file as a <TAB> separated flat text file, which can be opened in any text editor or spreadsheet processor such as MS Excel. Using the main menu item **Preferences→Regression Preferences** you can switch the output format, for example to the intervendor XML standard for data mining models, PMML.
- The currently specified settings will automatically be saved to an XML parameter file named `reg_params_customers.xml` every time the button **Start training** will be pressed. The resulting XML file can be reloaded in a later Synop Analyzer session via the main menu item **Analysis→Run regression analysis**. This reproduces exactly the currently active parameter settings and data import settings.
- As the target field we choose the field `AccountBalance`. Hence, we want to create a linear regression model which is able to predict the presumable account balance, for example for new customers.
- In the input field **max. regressor fields** one can limit the maximum number of predictor fields which may appear in the regression model. We do not enter a value here. If we had done it, Synop Analyzer would automatically select those predictor fields which have the maximum linear correlation with the target field.
- In the checkbox **Include constant offset term** you can specify whether or not the model can contain a constant offset c_0.
- By marking the checkbox **Replace missing predictor values by mean value** you can modify the treatment of missing predictor field values. Per default, a missing value of a numeric predictor field is assumed as 0, that means it has no impact in the predicted target field value. If the checkbox is marked, missing values are replaced

by the field's mean value when calculating the field's contribution to the target field value.

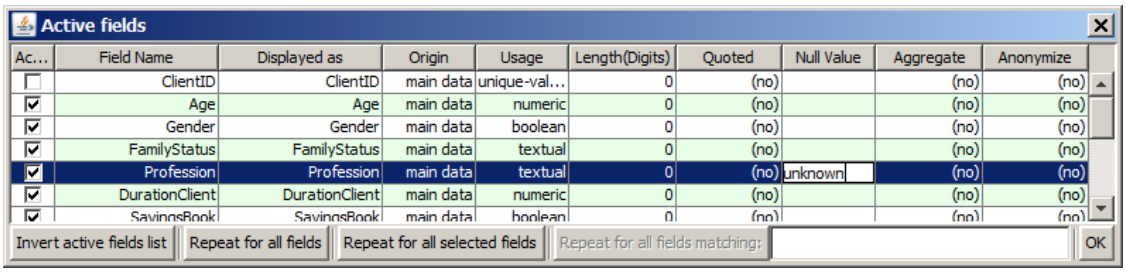
- If the checkbox **Create a new residual field in the data** is marked, a new data field will be appended to the training data at the end of the training process. The new field contains the residuals, that means the differences 'actual target field value minus predicted target field value'. This information can be helpful for judging the quality and usability of the model for the intended purposes. For example, you can examine in which situation and on which data records the model delivers a good prediction accuracy and in which cases it does not.

When working with the module Linear Regression Analysis, you sometimes get the following error message:


Nachricht
At least two of the regressors are collinear. No model can be built.
To avoid this problem, deactivate highly correlated data fields and correctly specify all field values representing unknown, missing or invalid values as 'Null values' in the 'Active fields' dialog.
OK

The message says that some of the predictor fields are collinear, that means perfectly correlated. In this case, no unambiguous linear regression model can be built. In the example mentioned above, the message appears due to the fact that the data field `Profession` contains a the value `unknown` on couple of data records, which all happen to have ages below 12 and the family status `child`. All other data records with these data subset have `Profession=inactive`. Therefore, the two profession values are collinear.

The problem can be resolved by defining the value `unknown` as a 'null value' for the field `Profession` within the pop-up dialog **Active data fields** before reading the data into memory. The effect of this is that the value `unknown` does not any more represent a valid field value, and no regressor is created for it.

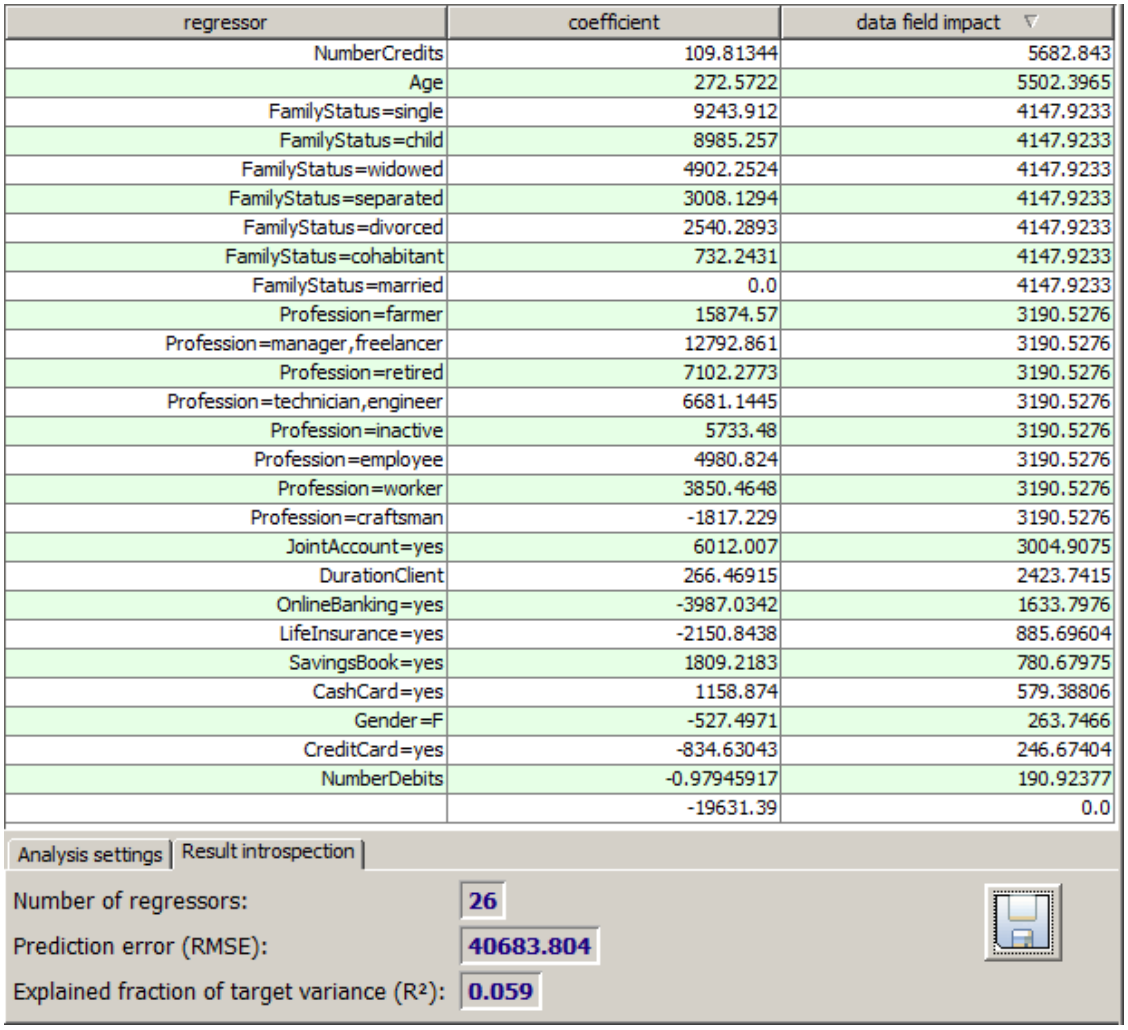

Active fields
Ac... Field Name Displayed as Origin Usage Length(Digits) Quoted Null Value Aggregate Anonymize
ClientID ClientID main data unique-val... 0 (no) (no) (no)
Age Age main data numeric 0 (no) (no) (no)
Gender Gender main data boolean 0 (no) (no) (no)
FamilyStatus FamilyStatus main data textual 0 (no) (no) (no)
Profession Profession main data textual 0 (no) unknown (no) (no)
DurationClient DurationClient main data numeric 0 (no) (no) (no)
Invert active fields list
Repeat for all fields
Repeat for all selected fields
Repeat for all fields matching:
OK

3.12.3 The Regression result panel

After successfully terminating the training process, the main part of the window 'Regression Analysis' displays the regressors of the resulting model and their coefficients c_i in the first two columns of the tabular result view.

The right column of the table ranks the corresponding predictor fields by their importance within the regression model, that means by their average impact on the predicted target

field values. For calculating this impact measure, the software memorizes for each data record the contribution to the target value which comes from all regressors deriving from the one single examined predictor field. The displayed number is then the standard deviation of this list of contribution numbers.



regressor	coefficient	data field impact
NumberCredits	109.81344	5682.843
Age	272.5722	5502.3965
FamilyStatus=single	9243.912	4147.9233
FamilyStatus=child	8985.257	4147.9233
FamilyStatus=widowed	4902.2524	4147.9233
FamilyStatus=separated	3008.1294	4147.9233
FamilyStatus=divorced	2540.2893	4147.9233
FamilyStatus=cohabitant	732.2431	4147.9233
FamilyStatus=married	0.0	4147.9233
Profession=farmer	15874.57	3190.5276
Profession=manager,freelancer	12792.861	3190.5276
Profession=retired	7102.2773	3190.5276
Profession=technician,engineer	6681.1445	3190.5276
Profession=inactive	5733.48	3190.5276
Profession=employee	4980.824	3190.5276
Profession=worker	3850.4648	3190.5276
Profession=craftsman	-1817.229	3190.5276
JointAccount=yes	6012.007	3004.9075
DurationClient	266.46915	2423.7415
OnlineBanking=yes	-3987.0342	1633.7976
LifeInsurance=yes	-2150.8438	885.69604
SavingsBook=yes	1809.2183	780.67975
CashCard=yes	1158.874	579.38806
Gender=F	-527.4971	263.7466
CreditCard=yes	-834.63043	246.67404
NumberDebits	-0.97945917	190.92377
	-19631.39	0.0

The tab **Result introspection** within the bottom tool bar displays the total number of regressors within the model, and it contains two quality numbers which help to judge the quality of the generated model:

- The **Prediction error (RMSE, 'root mean squared error')** is the standard deviation of the residual 'actual target field value minus predicted target field value' on the training data. Hence, the value describes the mean prediction accuracy.
- The measure **Explained fraction of variance** describes which fraction of the actually observed deviation of the single records' target field values from their mean value is correctly predicted by the model. A perfect model would have the value of 1, a random model, or a model which always predicts the mean target field value would have the value 0.

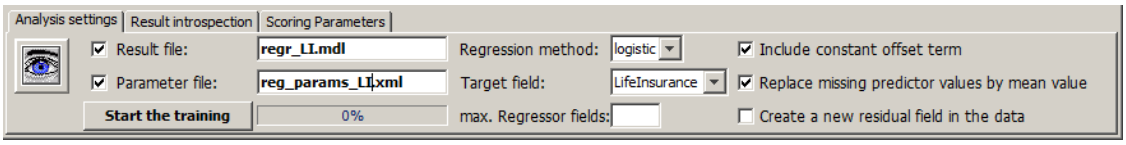
From the low quality values of our sample model we can see that linear regression models often deliver poor prediction quality. This is due to the fact that the linear regression approach is mathematically simple but completely neglects many important possible types of relations between the target field value and the predictor values. In particular, non-linear relations such as quadratic, exponential or cyclic relations can not be modeled, and the same holds for multi factor effects such as $y = c * x_i * x_j$.

Therefore, linear regression models should be used for actually predicting values ('scoring') with care. Rather, they are useful for studying the principal relations between different fields, and for serving as reference models for regression models created by more sophisticated algorithms such as SOM or regression trees.

3.12.4 Applying regression models to new data ('Scoring')

Regression models can be applied to new data in order to create predictions on these data. This application of regression models to new data for predictive purposes is called 'scoring'.

You load and apply a linear or logistic regression model by first opening and reading the new data, by then pressing the button in order to start the regression analysis module and by then clicking the button **Load model** in the tab **Scoring Settings** of the tool bar at the lower end of the panel's GUI window.



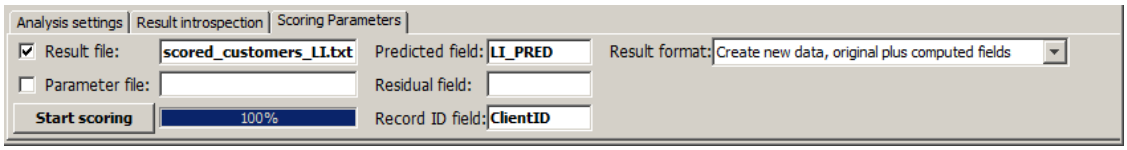
In the following sections we will demonstrate the process of regression model scoring with the help of a concrete example use case: using an logistic regression model we want to predict the propensity of newly acquired bank customers to sign a life insurance contract.

For this purpose, we load the sample data `doc/sample_data/customers.txt`. We keep the default data import settings with one exception but mark the field `CUSTOMER_ID` as the **group** field in the pop-up window **Active Fields**. Then we start the regression analysis module and train a model called `regr_li.mdl`, using the following parameter settings:

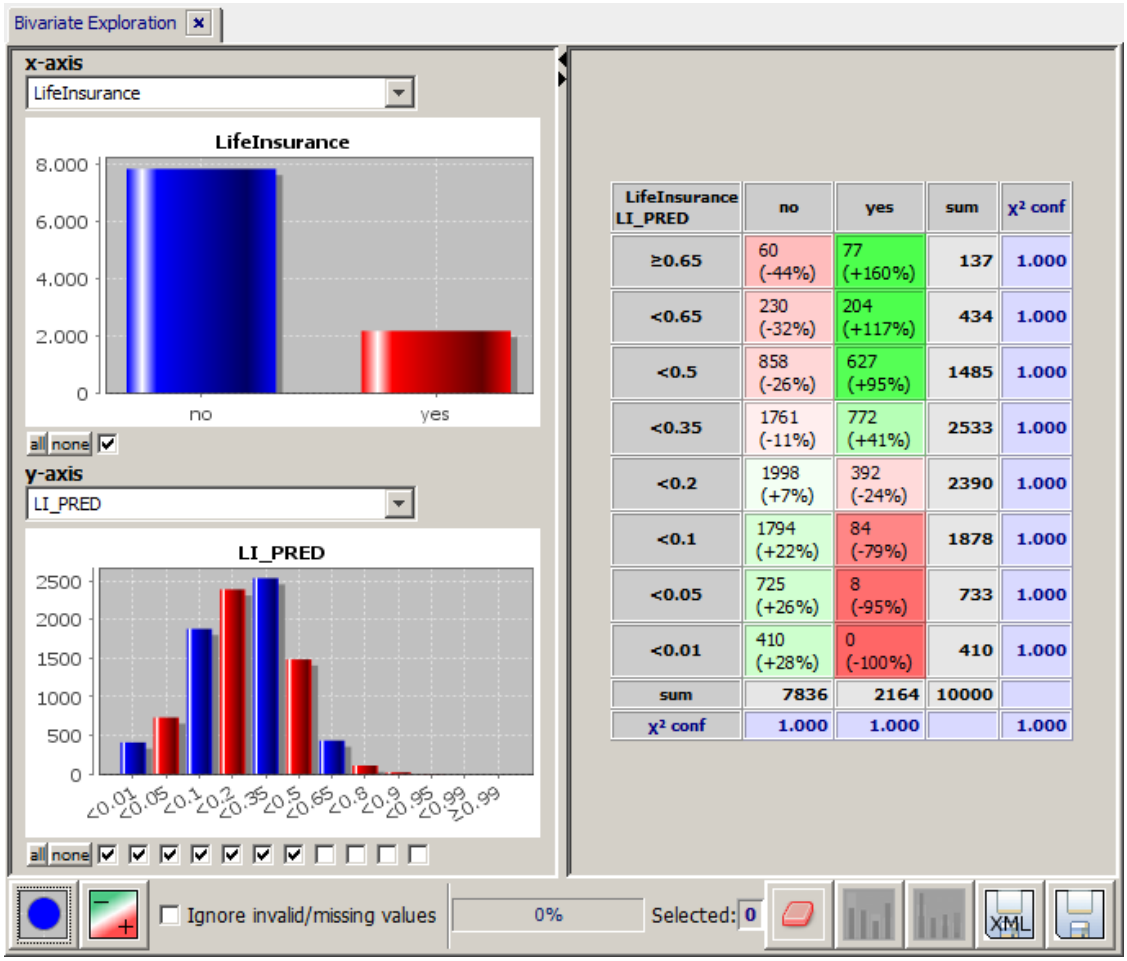
- Regression method **logistic**,
- Target field `LifeInsurance`,

For model evaluation purposes, we apply the generated model to the training data and compare the predicted life insurance propensity to the actual existence or non-existence of a life insurance contract. In the **Scoring Settings** tab of the toolbar, we specify that we want to create a predicted field called `LI_PRED`. Optionally, we can also specify a file

name to which the scoring results will be written (`scored_customers_LI.txt`). Then we press **Start scoring** in order to create the desired scoring results.

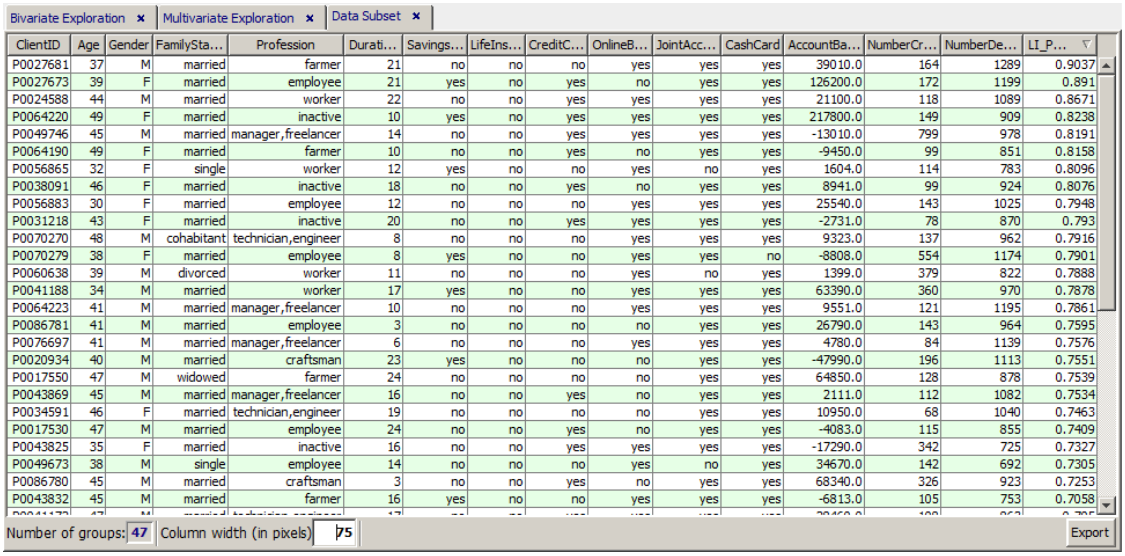


A new in-memory data source tab pops up in the left column of the Synop Analyzer workbench. In this new data source, we select the module **Bivariate Exploration** and select the data fields `LifeInsurance` as x-axis field and `LI_PRED` as y-axis field. The red and green colors in the bivariate matrix show us that the model's prediction generally coincides well with the actual values.



LifeInsurance / LI_PRED	no	yes	sum	χ^2 conf
≥0.65	60 (-44%)	77 (+160%)	137	1.000
<0.65	230 (-32%)	204 (+117%)	434	1.000
<0.5	858 (-26%)	627 (+95%)	1485	1.000
<0.35	1761 (-11%)	772 (+41%)	2533	1.000
<0.2	1998 (+7%)	392 (-24%)	2390	1.000
<0.1	1794 (+22%)	84 (-79%)	1878	1.000
<0.05	725 (+26%)	8 (-95%)	733	1.000
<0.01	410 (+28%)	0 (-100%)	410	1.000
sum	7836	2164	10000	
χ^2 conf	1.000	1.000		1.000

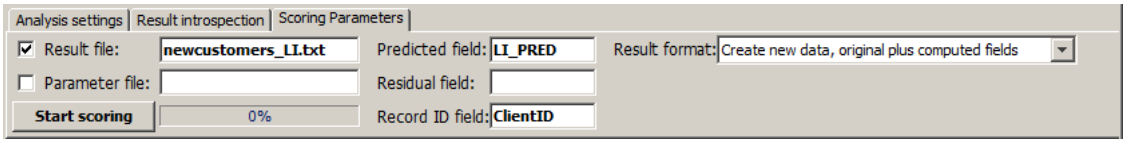
The predictions calculated on the training data could also be used for detecting interesting 'candidates' for sales actions concerning life insurance contracts. For example, one could select the customers under 50 years which do not (yet) have a life insurance but for which the model has predicted a high propensity for signing a life insurance contract:



ClientID	Age	Gender	FamilySta...	Profession	Durati...	Savings...	LifeIns...	CreditC...	OnlineB...	JointAcc...	CashCard	AccountBa...	NumberCr...	NumberDe...	LI_P...
P0027681	37	M	married	farmer	21	no	no	no	yes	yes	yes	39010.0	164	1289	0.9037
P0027673	39	F	married	employee	21	yes	no	yes	no	yes	yes	126200.0	172	1199	0.891
P0024588	44	M	married	worker	22	no	no	yes	yes	yes	yes	21100.0	118	1089	0.8671
P0064220	49	F	married	inactive	10	yes	no	yes	yes	yes	yes	217800.0	149	909	0.8238
P0049746	45	M	married	manager,freelancer	14	no	no	yes	yes	yes	yes	-13010.0	799	978	0.8191
P0064190	49	F	married	farmer	10	no	no	yes	no	yes	yes	-9450.0	99	851	0.8158
P0056865	32	F	single	worker	12	yes	no	no	yes	no	yes	1604.0	114	783	0.8096
P0038091	46	F	married	inactive	18	no	no	yes	no	yes	yes	8941.0	99	924	0.8076
P0056883	30	F	married	employee	12	no	no	no	yes	yes	yes	25540.0	143	1025	0.7948
P0031218	43	F	married	inactive	20	no	no	yes	yes	yes	yes	-2731.0	78	870	0.793
P0070270	48	M	cohabitant	technician,engineer	8	no	no	no	yes	yes	yes	9323.0	137	962	0.7916
P0070279	38	F	married	employee	8	yes	no	no	yes	yes	no	-8808.0	554	1174	0.7901
P0060638	39	M	divorced	worker	11	no	no	no	yes	no	yes	1399.0	379	822	0.7888
P0041188	34	M	married	worker	17	yes	no	no	yes	yes	yes	63390.0	360	970	0.7878
P0064223	41	M	married	manager,freelancer	10	no	no	no	yes	yes	yes	9551.0	121	1195	0.7861
P0086781	41	M	married	employee	3	no	no	no	no	yes	yes	26790.0	143	964	0.7595
P0076697	41	M	married	manager,freelancer	6	no	no	no	yes	yes	yes	4780.0	84	1139	0.7576
P0020934	40	M	married	craftsman	23	yes	no	no	no	yes	yes	-47990.0	196	1113	0.7551
P0017550	47	M	widowed	farmer	24	no	no	no	no	yes	yes	64850.0	128	878	0.7539
P0043869	45	M	married	manager,freelancer	16	no	no	yes	no	yes	yes	2111.0	112	1082	0.7534
P0034591	46	F	married	technician,engineer	19	no	no	no	no	yes	yes	10950.0	68	1040	0.7463
P0017530	47	M	married	employee	24	no	no	yes	no	yes	yes	-4083.0	115	855	0.7409
P0043825	35	F	married	inactive	16	no	no	yes	yes	yes	yes	-17290.0	342	725	0.7327
P0049673	38	M	single	employee	14	no	no	no	yes	no	yes	34670.0	142	692	0.7305
P0086780	45	M	married	craftsman	3	no	no	yes	no	yes	yes	68340.0	326	923	0.7253
P0043832	45	M	married	farmer	16	yes	no	no	yes	yes	yes	-6813.0	105	753	0.7058

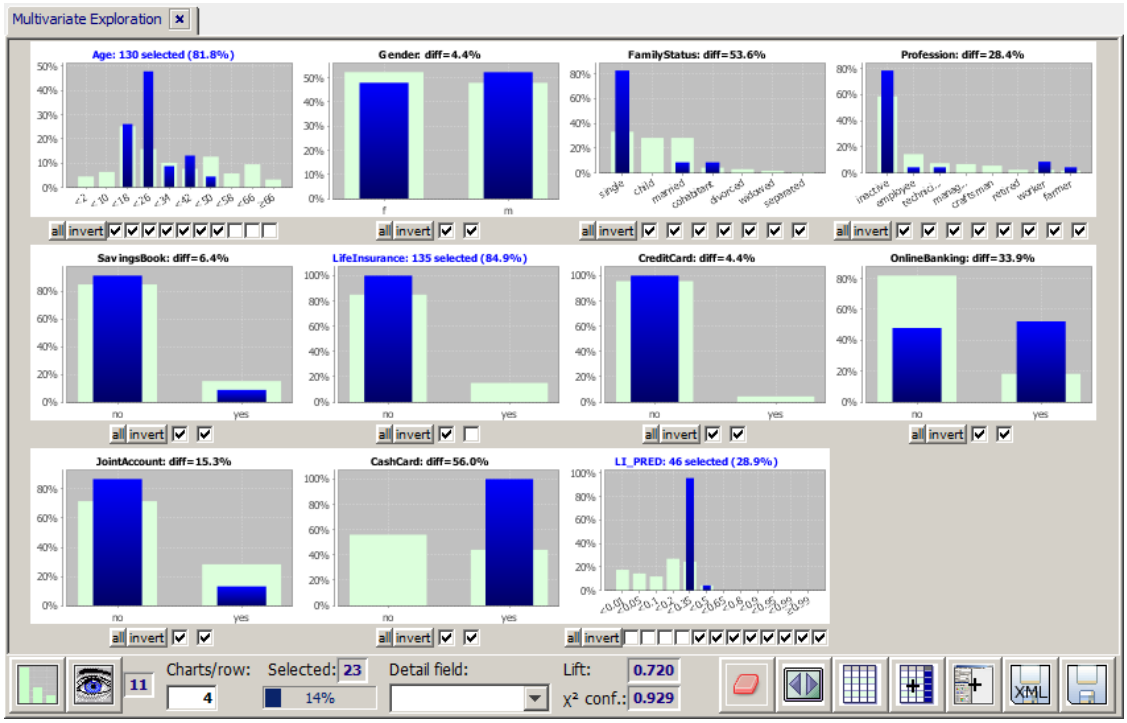
Now we apply the logistic regression model to a new data collection: 159 new customers, among which we want to find the most interesting candidates for selling life insurance contracts. We load the data `doc/sample_data/newcustomers_159.txt` into Synop Analyzer, thereby marking the data field `CUSTOMER_ID` as the 'group' field in the pop-up dialog **Active fields**

On this new in-memory data source, we start the regression analysis module and move to the tab **Scoring Parameters** in the tool bar at the lower end of the screen. Here, first load the regression model to be applied to the data, the model `regr_LI.mdl`. Then, we enter the name of the file in which the scoring results are to be stored (`newcustomers_LI.txt`), we define the scoring result data fields to be contained in that file and we specify that the new file should be a copy of the existing file `newcustomers_159.txt` plus the new computed data fields. (**Create new data, original plus computed fields**).



By means of the button **Start scoring** we create the scoring results, write the desired result file to disk and open the resulting data as a new in-memory data source in Synop Analyzer, that means as a new tab in the left column of the Synop Analyzer workbench.

We introspect the scoring result data with the module 'multivariate exploration'. We see that the model has created a propensity probability of at least 20% for 23 of the 159 new customers whose age is below 50 years and who do not yet have a life insurance.



Via the button we submit the selected 23 data records to a last visual examination. Then we can use the button **Export** to save the resulting list to a flat file or Excel spreadsheet, or we can use the main menu button **Report** to create a HTML or PDF report.

Multivariate Exploration | Data Subset

ClientID	Age	Gender	FamilyStatus	Profession	SavingsBook	LifeInsurance	CreditCard	OnlineBanking	JointAccount	CashCard	LI_PRED
P0196161	30	f	single	inactive	no	no	no	yes	no	yes	0.3619
P0197007	20	f	single	inactive	no	no	no	yes	no	yes	0.3212
P0197442	35	m	single	worker	no	no	no	no	no	yes	0.3177
P0197062	19	f	single	inactive	no	no	no	yes	no	yes	0.3173
P0195682	18	f	single	inactive	no	no	no	yes	no	yes	0.3134
P0196499	18	f	single	inactive	no	no	no	yes	no	yes	0.3134
P0196757	24	f	single	inactive	no	no	no	no	no	yes	0.3052
P0199933	17	f	single	inactive	yes	no	no	no	no	yes	0.2954
P0197192	21	f	single	inactive	no	no	no	no	no	yes	0.2938
P0196848	20	m	single	worker	no	no	no	yes	no	yes	0.2913
P0195839	18	m	single	inactive	no	no	no	yes	no	yes	0.2838
P0195686	17	m	single	inactive	no	no	no	yes	no	yes	0.2801
P0195969	17	m	single	inactive	no	no	no	yes	no	yes	0.2801
P0196538	16	m	single	inactive	no	no	no	yes	no	yes	0.2765
P0199925	16	f	single	inactive	no	no	no	no	no	yes	0.2754
P0196424	22	m	single	inactive	no	no	no	no	no	yes	0.2689
P0195966	17	m	single	inactive	yes	no	no	no	no	yes	0.2668
P0197536	43	f	married	employee	no	no	no	yes	yes	yes	0.2608
P0197085	25	f	cohabitant	inactive	no	no	no	yes	no	yes	0.26
P0196726	40	m	single	technician,e...	no	no	no	no	no	yes	0.2557
P0196951	18	m	single	inactive	no	no	no	no	no	yes	0.2549
P0197460	40	m	married	farmer	no	no	no	no	yes	yes	0.2282
P0197183	28	m	cohabitant	inactive	no	no	no	no	yes	yes	0.2201

Number of groups: 23 | Column width (in pixels): 75 | Export

4

XML API and Task Automization

In this part of the user's guide we describe the command line processor and Synop Analyzer's XML API, and we show how they can be used for creating automated data analytics workflows.

XML API: Based on a XML interface, Synop Analyzer can be used as an analysis kernel within automated workflows or batch processes, or as a plugin component embedded into third-party software.

Command Line Processor: Synop Analyzer can not only be used via a graphical Frontend (workbench) but also as a command line tool without graphical user interaction. The command line version of Synop Analyzer is called `sacl`. It is particularly suited for creating automated (batch) analysis workflows which run regulary without user interaction.

Reporting: This help page describes the features for visually designing or editing predefined report templates, and for executing these report templates on a given data source in order to obtain PDF or HTML reports on the most up-to-date data.

4.1 The XML Application Programming Interface

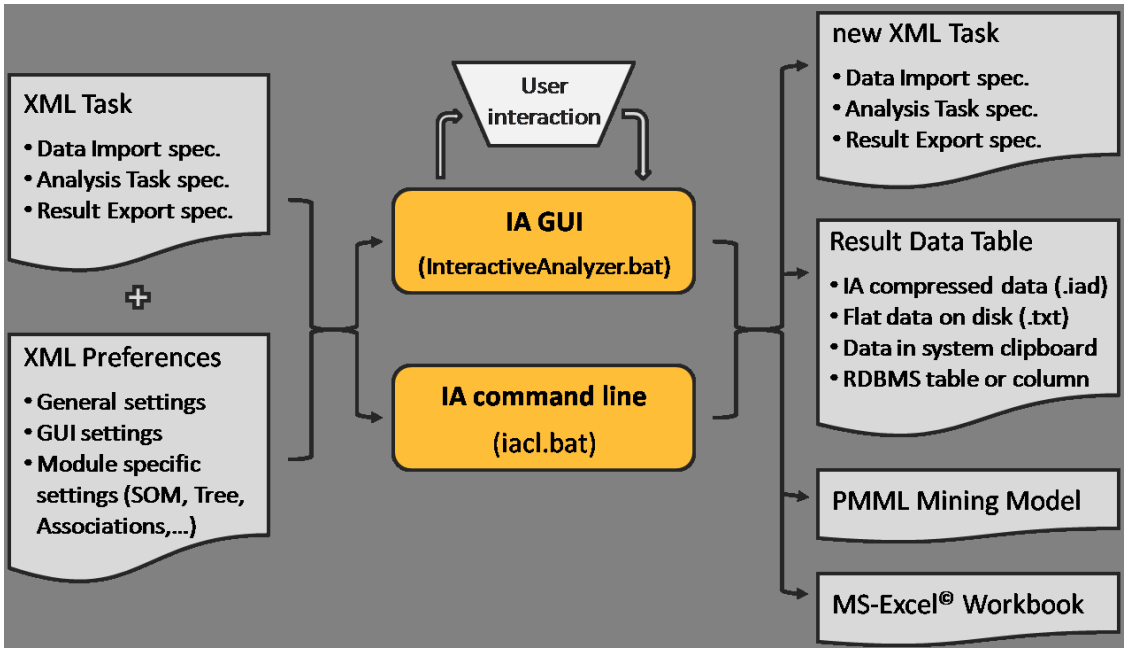
4.1.1 Command line parameters and the command line processor sacl

Based on a XML interface, Synop Analyzer can be used as an analysis kernel within automated workflows or batch processes, or as a plugin component embedded into third-party software. Synop Analyzer can be called in two ways

- as 'workbench' with graphical user interface (GUI) for working interactively(`SynopAnalyzer.bat`),
- as command line processor which processes a given analysis task without user interaction(`sacl.bat`).

The first calling variant can take, the second one must take 1 or 2 command line parameters:

- An analysis task in the form of an XML document which can be validated against the XML schema
  `http://www.synop-systems.com/xml/InteractiveAnalyzerTask.xsd`,
- The name of the XML file which contains the preference settings to be used. This file must validate against the XML schema:
  `http://www.synop-systems.com/xml/InteractiveAnalyzerPreferences.xsd`.



In the following sections of this document, syntax and usage of Synop Analyzer XML tasks will be described in more detail.

4.1.2 General structure and a simple example of an XML task

An XML task according to the XML schema http://www.synop-systems.com/xml/-InteractiveAnalyzerTask.xsd consists of two parts:

- a description of the data to be analyzed, in the form of an <InputData> tag,
- a description of the analysis task to be performed.

A simple task, which reads the flat file kunden.txt from the subdirectory doc/sample_-data of the Synop Analyzer installation directory and opens it in the Synop Analyzer graphical workbench, is given below:

```
<?xml version="1.0" ?>
<InteractiveAnalyzerTask>
   <InputData>
      <InputDataLocator usage="DATA_SOURCE" type="FLAT_FILE"
         name="doc/sample_data/kunden.txt"/>
   </InputData>
   <StartInteractiveAnalyzerGUITask/>
</InteractiveAnalyzerTask>
```

If you store this task as kunden_task1.xml and start SynopAnalyzer or sacl with this file name as command line argument,

```
c:> SynopAnalyzer.bat kunden_task1.xml,
```

then the Synop Analyzer workbench opens up, the data are automatically read into memory, and after a few seconds, you can start analyzing them. That means, the data from kunden.txt were read, interpreted, compressed, enriched with additional statistics and are now available in the computer's RAM for arbitrary analysis or data exploration tasks.

You could also submit the XML task directly as a textual string when calling Synop-Analyzer or sacl. In this case, however, you have to 'quote' the task by enclosing it into double quotes. The existing double quotes within the string have to be masked by backslashes (\) in this case. The call would then look like this:

```
c:> SynopAnalyzer.bat "<?xml version=\"1.0\" ?><InteractiveAnalyzerTask>
<InputData><InputDataLocator usage=\"DATA_SOURCE\" type=\"FLAT_FILE\"
name=\"doc/sample_data/kunden.txt\"/></InputData>
<StartInteractiveAnalyzerGUITask/></InteractiveAnalyzerTask>"
```

4.1.3 Reference description of the <InputData> part

The element <InputData> describes a data source which can be opened in Synop Analyzer

Optional attributes of <InputData>

- **nbThreads**: maximum number of parallel threads used while reading and compressing the input data. If this value is missing or smaller than 1, all available CPU cores will be used for spawning one separate threads per core.
- **nbDigits**: precision (number of digits) with which floating point numbers are stored in the compressed data format. For statistical analysis and Data Mining, rarely more than 4 digit precision is needed, hence 4 is the predefined value. This value can be increased up to a maximum of 8.
- **nbRecordsForDataDescription**: number of data rows which are read for detecting the most probable field types of data fields when reading flat file data. The default value is 1000.
- **maxNbCharacters**: long textual field values are truncated after a certain number of characters while reading and compressing the input data. The default value is 40.
- **maxNbNumericHistogramBins**: defines the level of detail in the histogram charts that are created for numeric data fields. The predefined value is 10, that means the histograms for numeric data fields have up to 10 histogram bars.
- **maxDiffTextualValues**: determines how many different textual values are stored in the compressed data representation of textual data fields. The most frequent values are stored separately, the remaining values are grouped into the category 'others'. Default value is 2000, that means the 2000 most frequent values of each textual data field are treated as separate values.
- **maxNbActiveFields**: if this value is smaller than the number of available data fields in the input data, Synop Analyzer automatically deactivates data fields until not more than `maxNbActiveFields` active data fields remain. During this removal process, each data field is ranked with respect to several criteria: number of missing values, number of different values, predominance of the most frequent value, existence of high correlations with other fields. The joint score of these criteria provides a 'field importance' score, and the fields with smallest scores are deactivated. Per default, this mechanism is switched off, all active fields are kept.
- **allowIrreversibleBinning**: if this attribute is set to `"true"`, numeric data fields are irreversibly 'binned' into `maxNbNumericHistogramBins` different value ranges (bins) if they initially contain more than `maxNbNumericHistogramBins` different values. This irreversible binning reduces the size of the compressed data. Per default, irreversible binning is switched off.
- **anonymizationLevel**: defines, whether and how strongly data field names and data field values are (irreversibly) anonymized when reading input data.
  0 (default): no anonymization,
  1: anonymize the field names, keep the original field values,
  2: anonymize the textual field values and transform all numeric field values such that the resulting value distribution for each numeric data field has a mean of 0 and

a standard deviation of 1. Maintain the original data field name,
3: anonymize both the data field name and the field values.

- **exportMode**: defines, whether and how the imported and preprocessed input data are to be stored persistently on disk. The following export formats are available:
  `"COMPRESSED_IAD"`: the data are stored in the proprietary Synop Analyzer Data Format (.iad), a compressed binary data format which consumes 5% to 10% of the original data size.
  `"PIVOTED"`: the data are stored in a two-column, pivotized form. One column contains the record ID (or, if a 'group' column has been specified, the group ID). The other column contains, in several adjacent data rows, all combinations `data field=value` which appear in the original data for the given record or group ID.
  `"SET_VALUED"`: writes uncompressed text data with one column per original data field. If no 'group' field has been defined then the exported data exactly correspond to the input data if these were read from a flat text file. If a 'group' field has been specified and in the original data one group ID can span several data rows, then the exported format is different: it always contains one single data rows for each group ID. If certain data fields have several values per group ID, the entire set of values is stored as one single textual string, enclosed by curly braces \\.
  `"BOOLEAN_FIELDS"`: transforms pivoted input data in a data format with a large number of two-valued (yes/no) data fields and exactly one data row per group ID. Each of the new data fields stands for one combination 'data field=value' from the original data, and this field contains the value "1" if the current groupID contains the combination 'data field=value', and "0" otherwise.
  `"GROUP_ID"`: writes a file which contains one single column. This column contains a record ID, or, if a 'group' field has been specified, the group ID. This data format is not useful for storing the entire data, but very helpful for storing previously selected data subsets, for example the customer IDs of previously selected customers etc.

<DataLocator> subelements

`<InputData>` must contain one and can contain two more different subelements of type `<DataLocator>`:

- **<InputDataLocator>**: contains the URL (access path and data name) and the data format of a data source which contains input data to be opened with Synop Analyzer.
- **<TaskDataLocator>**: if the user wants to permanently store the manual adjustments and data import settings performed on the current data source, `<TaskDataLocator>` specified the URL to which these settings are written in the form of an `<InteractiveAnalyzerTask>`.
- **<OutputDataLocator>**: if the user wants to permanently store the imported and preprocessed data source, the URL for this persistent data file must be given.

Each `<DataLocator>` must contain the following three attributes:

- **type**: describes the data type (format). Must be one of the following constants: "FLAT_FILE", "OOXML_SPREADSHEET", "COMPRESSED_IAD","XML_FILE", "PMML_-FILE", "JDBC_TABLE", or "MDB_TABLE"
- **usage**: describes the data usage. Must be one of the following constants: "DATA_-SOURCE", "DATA_TARGET", "IA_PARAMETERS", or "IA_MODEL"
- **name**: datafile name or schema and table name

Optionally, <DataLocator> can contain one or more of the following attributes

- **accesspath**: directory path or JDBC connection string (containing DBMS, server, port and database name)
- **encoding**: encoding scheme of the data source. Allowed values are
  "US-ASCII": suitable if the data only contains the first 127 characters of the ASCII table.
  "ISO-8859-1": for 'western European' data in which each character is represented by one single byte and in which the 127 ASCII characters plus some 'standard' western European characters (such as French accents or German Umlauts) occur.
  "ISO-8859-15": codepage specialized for German language information. Each character is represented by one single byte and the 127 first characters are the ASCII characters, the other 128 characters represent characters and other symbols which are frequently used in the German speaking countries (Germany, Austria, Switzerland).
  "UTF-8": The UTF coding standard can represent about 65000 regionally used characters from all over the world. In the variant UTF-8, the first 127 ASCII characters are represented by 1 byte, all other characters are represented by two or more bytes.
  "UTF-16": in the UTF-16 variant of the UTF standard, all characters are represented by two or more bytes. The first two bytes of an UTF-16 file contain information on the byte order (is the first byte the high byte or the low byte?)
  "UTF-16LE", "UTF-16BE": in the variants UTF-16LE ('little endian') and UTF-16BE ('big endian'), the first two bytes of the UTF-16 format which define the byte order are missing. Therefore, the user must know beforehand, which byte order the creator of the document used. Normally, Intel/Windows systems work with the 'little endian' convention, Unix systems and Mainframes in the 'big endian' convention.
  "ISO-8859-2", "ISO-8859-4", "ISO-8859-5", "ISO-8859-7", "ISO-8859-9", "ISO-8859-13", "KOI8-R", "windows-1250", "windows-1251", "windows-1252", "windows-1253", "windows-1254", "windows-1257": other possible codepages which will not be described in detail here.
- **jdbcUser**: database user name
- **dbms**: name of the database management system in which the data reside. Possible values are "ORACLE", "SQLSERVER", "ACCESS", "DB2", "MYSQL", "POSTGRES", "SYBASE", "TERADATA", "PROGRESS", CACHE, "SUN_ODBC_JDBC", "USERDEFINED" or "NONE", the latter being the default value.

Optional <FieldUsage> subelements

<FieldUsage> defines a usage specification for one single data field. The tag contains the following attributes:

- **field**: name of the data field (required).
- **alias**: mapped name of the data field (optional). This mapped name is used instead of the field name in captions and titles of histogram charts for the field.
- **dataType** defines the data type class of the data field: `"DEFAULT"`, `"TEXTUAL"`, `"BOOLEAN"`, `"INTEGER"`, or `"NUMERIC"`. If this attribute is not set, `"DEFAULT"` is assumed. That means, Synop Analyzer autonomously detects the best matching data type class for that data field.
- **usage**: usage mode of the field in all data exploration and analysis steps to be performed on this data.
  `"SUPPRESSED"`: the field will be ignored.
  `"SUPPLEMENTARY"` (this usage type is not used in Synop Analyzer v1.x).
  `"ACTIVE"`: the default usage type.
  `"GROUP"`: the field is the 'group' field: it contains group IDs which mark a group of adjacent data rows as members of one group.
  `"ENTITY"`: the field is the 'entity' field: it contains a second grouping level on top of the 'group' field. The entity field contains entity IDs which marks a set of adjacent data row groups as members of one entity,
  `"WEIGHT"`: the field contains the weight, price or cost value which is associated with the situation, event or good described by the other data field values of the data record.
  `"ORDER"`: the field contains a time stamp or a date.
  If the `usage` attribute is not set, `"ACTIVE"` is assumed.
- **aggregationType** defines the value aggregation type. This attribute is only of interest for numeric data fields and for the case that a group field has been defined. The attribute determines how the field's values in different data records within one data group are aggregated in order to form the data group's value for that field.
  `"SUM"`: The field value of the data group (transaction) is the sum of the field values of all data records which form the group.
  `"MEAN"`: The field value of the data group is the average of the field values of all data records which form the group.
  `"MAX"`: The field value of the data group is the maximum of the field values of all data records which form the group.
  `"MIN"`: The field value of the data group is the minimum of the field values of all data records which form the group.
  `"SPREAD"`: The field value of the data group is the difference between the greatest and the smallest value of the field on all data records which form the group.
  `"RELATIVESPREAD"`: The field value of the data group is the difference between the greatest and the smallest value of the field on all data records which form the group divided by the mean field value on the data group.

`"MINDIFF"`: The field value of the data group is the minimum of all field value differences between two adjacent data records within the group.
`"MAXDIFF"`: The field value of the data group is the maximum of all field value differences between two adjacent data records within the group.
`"COUNT"`: The field value of the data group is the number of records which form the group.
The default aggregation type is `"SUM"`.

- **anonymizationLevel**: overrides the general anonymization level (defined as attribute `anonymizationLevel` of the `<InputData>` tag) for a single field:
0 (default): no anonymization,
1: anonymize the field names, keep the original field values,
2: anonymize the textual field values and transform all numeric field values such that the resulting value distribution for each numeric data field has a mean of 0 and a standard deviation of 1. Maintain the original data field name,
3: anonymize both the data field name and the field values.

- **dateFormat**: specifies the current field as a date/time field and indicates the date/-time format of the field, e.g. `"MM/dd/yyyy hh:mm:ss"`.

Optional <JoinedTable> subelements

<JoinedTable> specifies an auxiliary table which is to be combined with the main input data table using a primary key - foreign key relation between certain data fields of the two tables. `<JoinedTable>` has the following required attributes:

- **<DataLocator .../>**: URL and data format of the auxiliary table. See here for a detailed description of the element <DataLocator>.

- **<KeyFieldPair mainTableField="..." joinedTableField="..."/>**:
a pair of data fields, one from the main table, one from the auxiliary table, which serve as foreign key - primary key pair and thereby establish the relation between the two tables.

- **<AddedField field="..."/>**: the name of a data field from the auxiliary table which is to be added to the main table.

Optional <Taxonomy> subelements

<Taxonomy> defines an auxiliary table which contains taxonomy (hierarchy) information for one or more data fields of the main input data table. `<Taxonomy>` has the following required attributes:

- **parentField**: name of the data field in the auxiliary table which contains the 'parent', i.e. the higher order hierarchy level, of a taxonomy relation (parent-child relation).

- **childField**: name of the data field in the auxiliary table which contains the 'child', i.e. the lower order hierarchy level, of a taxonomy relation (parent-child relation).

`<Taxonomy>` must contain at least one of each of the following sub-tags:

- **<DataLocator .../>**: URL and data format of the auxiliary taxonomy table. The internal structure of this element has been described here.
- **<AffectedField field="..."/>**: a data field in the main data table for which the taxonomy relations apply.

Optional <NameMapping> subelements

<NameMapping> defines an auxiliary table which contains clear names for the values of one or more data fields of the main input data table. `<NameMapping>` has the following required attributes:

- **origNameField**: name of the data field in the auxiliary table which contains the original field values for which clear names are to be defined.
- **mappedNameField**: name of the data field in the auxiliary table which contains the mapped values (clear names).

`<NameMapping>` must contain at least one of each of the following sub-tags:

- **<DataLocator .../>**: URL and data format of the auxiliary name mapping table. The internal structure of this element has been described here.
- **<AffectedField field="..."/>**: a data field in the main data table for which the name mappings apply.

Optional <Discretization> subelements

<Discretization> describes a manually defined discretization (binning) for one or more data fields in the main input data table. `<Discretization>` can contain one integer valued numeric attribute:

- **nbBins**: number of intervals (bins), not counting a possible needed extra bin for invalid or missing values.

`<Discretization>` has the following sub-tags:

- **<BinBounds>...(StringList)...</BinBounds>**: the interval boundaries. This sub-tag is optional and only allowed if the discretization is defined for a numeric data field.
- **<AffectedField field="..."/>**: one data field in the main data table for which the discretization applies.

Optional <PerfectTupelDetection> subelements

<PerfectTupelDetection> defines a data analysis and data simplification step for data with set-valued data fields or data with a 'group' field. A perfect tupel detections identifies in a first step all combinations values of one data field which (almost) always occur together - that means in the same data records or data groups. In a second step, all values figuring in such a combination are removed from the data and replaced by a textual string representing the entire combination.`<PerfectTupelDetection>` can contain the following attributes:

- **minFrequency**: minimum frequency threshold for the perfect tupels to be detected. Default value is 10.
- **minPurity**: minimum purity threshold for the perfect tupels to be detected. Default value is 1.0, which means that only those tupels are detected and removed whose values never occur without all the other values from the tupel.
- **collationString**: text fragment or character which is used as link when composing the name of the combined tupel. Default value is '_'.

4.1.4 Reference description of the analysis task part

After the `<InputData>` part, an Synop Analyzer task can contain one or more of the following elements, which define various analysis tasks that can be performed on the data using Synop Analyzer different analysis modules.

- `<StartInteractiveAnalyzerGUITask>` (*)
- `<UnivariateExplorationTask>`
- `<CorrelationsTask>`
- `<BivariateExplorationTask>`
- `<MultivariateExplorationTask>`
- `<TestControlAnalysisTask>`
- `<TimeSeriesTask>`
- `<AssociationsTrainTask>`
- `<SequencesTrainTask>`
- `<RegressionTrainTask>`
- `<SOMTrainTask>`

The element marked with (*) is a 'dummy' element. It does not define an analysis step but just starts the Synop Analyzer workbench and reads the input data which habe been specified in the preceding `<InputData>` part of the task. This element can not be processed by the command line processor `iacl`.

<UnivariateExplorationTask>

<UnivariateExplorationTask> generates a statistical overview of the currently active input data and creates visualizations of the value distributions for all data fields.

`<UnivariateExplorationTask>` can contain the following attributes:

- **nbChartsPerRow**: number of field value distribution histograms shown in one row on screen. The higher the value, the smaller the size of each single histogram chart.
- **yAxisLabel**: a label text to appear next to the y axis of the histogram charts in the Univariate Exploration panel.
- **barColors**: a series of RGB color byte triples, such as 0:0:255 for the color blue, separated by blancs. The first triple defines the color of the first bar in each histogram, the second triple defines the second bar, and so on.

`<UnivariateExplorationTask>` can contain the following sub-elements:

- **<HiddenField field="..."/>** specifies a data field which is to be ignored in the statistical and visual data overview schreen. Note that data fields which have been marked with the `<FieldUsage usage="SUPPRESSED"/>` tag in the `<InputData>` element are ignored by default.
- **<ResultDataLocator .../>** defines name, access path and data format of the file or database table into which the result of the univariate exploration is to be exported. The internal structure of this element has been described in subsection `<DataLocator>`.

<CorrelationsTask>

<CorrelationsTask> analyses and displays correlations between the data fields.

`<CorrelationsTask>` can contain the following attributes:

- **minCorrelation** defines a lower limit for the correlation coefficients to be shown in the panel. The value must be in the range from 0.0 to 1.0.
- **field1**: if this attribute is set and contains a valid field name, only correlation coefficients involving that field are shown on screen.
- **field2**: if this attribute is set in addition to the attribute `field1`, then only the correlation coefficient between `field1` and `field2` is shown on screen.

`<CorrelationsTask>` can contain the following sub-element:

- **<ResultDataLocator .../>** defines name, access path and data format of the file or database table into which the result of the correlations analysis is to be exported. The internal structure of this element has been described in subsection `<DataLocator>`.

<BivariateExplorationTask>

<BivariateExplorationTask> creates a bivariate analysis of the interdependencies of two data fields. The values or value ranges of one field are traced along the x axis, the values of the second field along the y axis. The resulting matrix contains in each matrix cell (m,n) the number of data records - or, if a 'group' field has been specified, the number of groups - in which the x-field has the the m-th value and the y-field the n-th value. A color code signals whether this combination occurs more (green) or less (red) frequently than expected. This method visualizes systematic interdependencies between certain values of the two fields.

`<BivariateExplorationTask>` can contain the following attributes:

- **ignoreMissingValues**: if this value is set to 'true', all data records (respectively all groups) in which one of the two involved data fields has no valid value are ignored in the counts shown in the matrix cells. The default setting for `<ignoreMissingValues>` is 'false'.

- **showCirclePlot**: indicate whether or not to show an absolute frequency plot. If this attribute is missing, the plot is shown.

`<BivariateExplorationTask>` must contain the following required sub-elements:

- **<XField field="..." nbRanges="...">**
  <RangeBounds>...</RangeBounds>
  </XField>
  defines the x-axis field and its binning into diskrete ranges. Each discrete range corresponds to one column in the resulting bivariate counts matrix.
  `nbRanges` is the number of ranges (columns); the sub-element `RangeBounds` contains a series of digits 1 and 0, separated by blancs. The series must contain `nbRanges`-1 times the digit 1 and in total n-1 digits, where n is the number of different field values (or discretized ranges as defined in the `InputData` part of the XML task).
  Example: we assume that the data field `AGE` is a numeric data field with more than 10 different values, and no `<Discretization>` has been specified for this field in the `<InputData>`part of the task. Then `AGE` will be discretized into 10 value ranges (bins), plus an additional 11-th range 'missing/invalid' if the field contains missing or invalid values. Hence, `<RangeBounds>` must contain 9 respectively 10 digits. It might look like this: `<RangeBounds>0 1 0 0 0 1 0 0 0</RangeBounds>`. In this case, the bivariate matrix has 3 columns. The first column represents the first two discrete bins of field `AGE`, the second column the next four bins and the last one the remaining four bins.

- **<YField field="..." nbRanges="...">**
  <RangeBounds>...</RangeBounds>
  </YField>
  defines the y-axis field and its binning into diskrete ranges. Each discrete range corresponds to one row in the resulting bivariate counts matrix.

- **<ResultDataLocator ... />**defines name, access path and data format of the file or database table into which the result of the bivariate exploration is to be exported. The internal structure of this element has been described in subsection <DataLocator>.

<MultivariateExplorationTask>

<MultivariateExplorationTask> generates and visualizes a multivariate data selection, that means the equivalent of a SQL SELECT statement with a WHERE clause in which one or more data fields appear as filter criteria. As a result, the multivariate selection shows how the value distributions of all data fields - the ones serving as selection criteria and the other ones - on the selected data subset differ from the corresponding value distributions on the entire data.

<MultivariateExplorationTask> can contain the following attribute:

- **nbChartsPerRow**: number of field value distribution histograms shown in one row on screen. The higher the value, the smaller the size of each single histogram chart.

<MultivariateExplorationTask> can contain the following sub-elements:

- **FieldHistogram field="..." nbBins="...">**
  <SelectedBins> ... </SelectedBins>
  </FieldHistogram>
  defines a selection criterion for the data field field.
  nbBins is the number of different values (or value ranges as defined in <InputData>).
  <SelectedBins> contains a series of digits 0 or 1, separated by blancs. The series must contain exactly nbBins digits. 1 signifies that the corresponding field value or value range is selected, 0 means that is is deselected.

- **<ResultDataLocator ... />** defines name, access path and data format of the file or database table into which the result of the multivariate exploration is to be exported. The internal structure of this element has been described in subsection <DataLocator>.

<TestControlAnalysisTask>

<TestControlAnalysisTask> creates and compares two different (and normally disjunct) multivariate data selections on one single data set: the 'test' data and the 'control' data. The two subsets can then be analyzed for significant value distribution differences. Furthermore, the test/control analysis module can sample a subset of the original control data which is 'representative' for the test data on some specified data fields.

<TestControlAnalysisTask> can contain the following optional attributes:

- **nbChartsPerRow**: number of field value distribution histograms shown in one row on screen. The higher the value, the smaller the size of each single histogram chart.
- **minNbControl**: minimum number of control data records (or groups) after sampling.
- **maxNbControl**: maximum number of control data records (or groups) after sampling.
- **iterateOverValuesOf**: name of a data field. This field will be used to define a series of test/control analysis tasks. Each task within the series defines one single value of the field as test data selector criterion and a set of other values of the field as the control data selector criterion.
- **maxNbIterations**: sets an upper limit for the number of different test/control analysis tasks generated by the attribute `iterateOverValuesOf`.
- **minChiSquareConfidence**: if this attribute is set (to a value beetween 0 and 1), test/control analysis results will only be generated and exported for those test/-control data splits with a certain minimum difference in the value distributions of at least one of the specified target fields. The value distribution is measured by a χ^2 test with the null hypothesis 'the value distributions of the test data and the control data for the target field are identical'.
- **summaryResultFile**: name of a 'summary' file which contains one line of data for each single test/control analysis within a series of automatically executed test/-control analysis steps. If this attribute is missing, no summary file will be written. If the name ends with 'xlsx', an Excel spreadsheet will be written, otherwise a tab-separated flat text file will be created.

`<TestControlAnalysisTask>` can contain the following sub-elements:

- **<FieldHistogramTC field="..." nbBins="..." optimizable="..."/>**
  specifies how the data field `field` is used within the test/control data analysis. `nbBins` is the number of different values (or value ranges as defined in `<InputData>`). `optimizable` indicates whether this field's value distribution on the control data should be made representative for the field's value distribution on the test data when the control data is being 'optimized'. Default value is 'true'.
  Furthermore, `<FieldHistogramTC>` can contain sub-elements which describe how the field is used as a splitting criterion for test and control data. Either
  `<SelectedBinsTest>...0 1 ...</SelectedBinsTest>`
  `<SelectedBinsControl>...0 1 ...</SelectedBinsControl>`,
  (if the selection criteria for the test and the control data are intended to differ on this field), or
  `<SelectedBins>...0 1 ...</SelectedBins>`
  (if identical selection criteria for both data sets are to be defined).
  Each `<SelectedBins...>` tag contains a series of digits 0 or 1, separated by blancs. The series must contain exactly `nbBins` digits. 1 signifies that the corresponding field value or value range is selected, 0 means that is is deselected.

- **<ResultDataLocator .../>** defines name, access path and data format of the file or database table into which the result of the split analysis is to be exported. The internal structure of this element has been described in subsection <DataLocator>.

<TimeSeriesAnalysisTask>

<TimeSeriesTask> describes the analysis of a time series: detection of trends and cyclic components ('seasons'), modeling the impacts of singular events ('strokes') and calculation of forecasts.

<TimeSeriesTask> can contain the following optional attributes:

- **nbChartsPerRow**: number of detail time series charts per row in the graphical overview. The larger the value, the smaller each single chart.
- **heightWidthRatio**: height-to-width ratio of the time series charts to be created.
- **groupingField**: name of the data field whose values are used to define the different detail time series and detail charts. For each value or value range of this field, a separate time series will be created and analyzed.
- **nbForecasts**: number of time steps to be predicted.
- **forecastStart**: time stamp at which the aggregation of aggregated forecast values (such as the forecasted total sales from January 1 till year end should start.
- **chartStart**: time stamp at which the generated charts should start.
- **exponentialSmoothingWeight**: weight factor (between 0.0 and 1.0) with which singular effects - strokes, deviations from the expected (trend+season) pattern - are influencing the prediction of future values.
- **exponentialSmoothingAlpha**: damping factor ($0.0 < \alpha < 1.0$; 0.0 means no damping) for the influence of deviations from the long-term (trend+season) pattern which happened in the recent past. A damping factor of α means that the influence of a deviation which happened `n` time steps ago will be damped by a factor of (1-α)ⁿ.
- **trendDamping**: trend damping factor `d > 0.0` models the expected behavior of the seasonally corrected trend line in the future. `d < (>) 1.0` assumes that the seasonally corrected trend which was detected in the recent past will be reduced (increased) by a factor of `d` with each time step into the future.
- **period**: presumed cycle length of the longest significant cyclic pattern (season) in the time series data. For example 12 if we assume a yearly pattern on monthly recorded data.
- **smoothing**: sliding average width. For calculating the seasonally corrected trend line, we use a symmetric sliding average over `smoothing` time steps. Default value is the value of `period`.

- **season**: defines the way in which the cyclic (seasonal) component is modeled into the data. Possible values are `ADDITIVE` and `MULTIPLICATIVE`. The first variant models the seasonal components as an additive contribution (added value), the second variant models it as a multiplicative factor (multiplication coefficient).
- **allowNegativeValues**: defines whether the forecast can contain negative values. The default value of this parameteris true.

`<TimeSeriesTask>` can contain the following subelement:

- **<ResultDataLocator ... />**: defines name, access path and data format of the file or database table into which the result of the time series analysis is to be exported. The internal structure of this element has been described in subsection `<DataLocator>`.

<AssociationsTrainTask>

<AssociationsTrainTask> defines the task to perform an associations analysis and to generate a collection of association rules on the data described in the `<InputData>` section. The result can be returned in the form of a PMML `<AssociationModel>` or in tabular form as a flat file.

`<AssociationsTrainTask>` can contain the following optional attributes:

- **nbVerificationRuns**: number of control models, which are calculated with the same rule filter settings as the main model but on artificially shuffled (permuted) data in which each data field's values are randomly moved to new data rows. By analyzing the significance numbers (support, lift, confidence) of the best 'artificial' rules detected on the control models, Synop Analyzer derives a reliability criterion for the associations and rules of the main model. This helps in differentiating true and robust patterns from artificial noise.
- **maxNbPatterns**: maximum number of associations to be detected. If more associations matchig all specified filter criteria can be found, they are sorted with respect to `sortingCriterion`, and only the best `maxNbPatterns` associations are kept.
- **sortingCriterion**: sorting criterion used for selecting the 'best' `maxNbPatterns` associations nach dem die 'besten' maxNbPatterns. Possible values are `"SUPPORT"`, `"LIFT"`, `"CONFIDENCE"`, `"PURITY"`, `"COREITEMPURITY"` und `"WEIGHT"`. Default value is `"SUPPORT"`.
- **minChildSupportRatio**: number between 0.0 and 1.0, with default value 0.0;
- **minParentSupportRatio**: number equal to or greater than 1.0, default value is 1.0; filter criterion which regulates the acceptance of short associations: an association of length `n-1` will only be accepted if its support (frequency of occurrence) is not smaller than `minParentSupportRatio` times the minimum of the supports of all 'child associations' of length `n` in which exactly 1 item is appended to the existing association. Setting `minParentSupportRatio` to a value greater than 1.0,

for example to 1.2, helps suppressing the appearance of masses of redundant partial patterns of one single interesting long pattern.

- **minChiSqrConf**: number between 0.0 and 1.0 with default value 0.0; if a value greater than 0.0 is set, for example 0.95, each detected association is submitted to a χ^2 significance test with the null hypothesis 'the appearance probability on the training data of at least one item within the association is independent of whether or not also the other `n-1` items of the association appear in the same data groups. The association will only be accepted if this null hypothesis is rejected at a significance level of at least `minChiSqrConf`. In other words: all associations are rejected in which at least one item seems not to be 'significant' for the entire pattern because its appearence probability is independent of the rest of the pattern.

`<AssociationsTrainTask>` can contain the following optional subelements:

- **<PatternLength min="..." max="..."/>**: lower and upper limit for the number of parts ('items') in the associations to be detected.
- **<AbsoluteSupport min="..." max="..."/>**: lower and upper limit for the absolute support (that is the absolute occurrence frequency on the training data) of the associations to be detected. Both limits must be integers greater than 0.
- **<RelativeSupport min=max="..."/>**: lower and upper limit for the relative support (that is the occurrence frequency on the training data divided by the total number of data groups in the training data) of the associations to be detected. Both numbers must be probability numbers between 0.0 (exclusive) and 1.0 (inclusive).
- **<RelativeItemSupport min="..." max="..."/>**: lower and upper limit for the relative supports of the single items which can occur in the associations to be detected. Both limits must be probability numbers between 0.0 (inclusive) and 1.0 (inclusive).
- **<Lift min="..." max="..."/>**: lower and upper limit for the lift of the associations to be detected. The lift of an association (A,B) is the relative support of (A,B) divided by the product of the relative supports of A and of B. Lift values greater (smaller) than 1 indicate a positive (negative) correlation between the items A and B. Both limits must be floating point numbers greater than 0.0.
- **<LiftIncreaseFactor min="..." max="..."/>**: lower and upper limit for the permitted lift ratios which result from comparing the lift of a 'child' association of length `n` to the lifts of all its 'parent' associations of length `n-1`. `<LiftIncrease-Factor>` greater than 1 enforces that only those items can be appended to existing parent patterns which have a positive correlation with the existing pattern. Both limits must be positive numbers.
- **<Purity min="..." max="..."/>**: lower and upper limits for the purity (on the training data) of the associations to be detected. Purity is a number between 0.0 and 1.0. In associations with purity 1.0, each single item within the association appears only in those data records or data groups in which also all other items of the association occur. More general, purity is defined as the support of the association

divided by the maximum of the supports of its items. Both limits must be numbers between 0.0 and 1.0.

- **<CoreItemPurity min="..." max="..."/>**: lower and upper limits for the core item purity (on the training data) of the associations to be detected. Core item purity is a number between 0.0 and 1.0; it is defined as the support of an association divided by the minimum of the supports of the association's items. In associations with a core item purity of 1.0, there is at least one item only occurs (on the training data) together with all other items of the association. Both limits must be numbers between 0.0 and 1.0.

- **<ItemPairPurity min="..." max="..."/>**: lower and upper limit for the pairwise purity of the items which are allowed to occur in the detected associations. Both limits must be numbers between 0.0 and 1.0. Setting a maximum item pair purity below 1.0 can be a means for suppressing the occurrence of well-known and trivial item-item correlations in the detected associations, for example combinations such as 'AGE<18' and 'MARITAL_STATUS=child'.

- **<Confidence min="..." max="..."/>**: lower and upper limits for the confidences of the 'if-then' rules which can be formed from the detected associations by taking one item as the 'then' part and all other items as the 'if' part of the rule. If this filter has been set, only those associations will be contained in the resulting model for which the confidence of at least one 'if-then' rule is in the specified range. Both limits must be probability numbers between 0.0 (exclusive) and 1.0 (inclusive).

- **<Weight min="..." max="..."/>**: lower and upper limit for the mean weights (prices or costs on the training data) of the associations to be detected. This filter criterion will be ignored unless a WEIGHT data field has been specified in the `<InputData>` section of the task. Both limits can be arbitrary numbers.

- **<RequiredItemGroups>**
  <ItemGroup><item>...</item> ...</ItemGroup>
  ...
  <ItemGroup><item>...</item> ...</ItemGroup>
  </RequiredItemGroups>
  defines bits of information ('items') which must occur in each association to be detected: from each `<ItemGroup>`, at least one item must occur in the patterns to be detected.

- **<IncompatibleItemGroups>**
  <ItemGroup><item>...</item> ...</ItemGroup>
  ...
  <ItemGroup><item>...</item> ...</ItemGroup>
  </IncompatibleItemGroups>
  defines bits of information ('items') which must not occur together in the patterns to be detected: from each `<ItemGroup>`, not more than one `<item>` may occur. Defining incompatible item groups is a means for eliminating the appearance of well known and trivial correlations from the detected associations.

- **<NegativeItems><item>...</item>...</NegativeItems>**:
  defines those bits of information ('items') for which not only the appearance but also the non-appearance within a data record or data group can become part of a detected pattern.
- **<SuppressedItems><item>...</item> ...</SuppressedItems>**:
  defines those bits of information ('items') which are to be completely ignored during the associations analysis.
- **<TrackedItems><item>...</item> ...</TrackedItems>**:
  defines certain bits of information ('items') for which the relative occurrence frequency (relative support) on the support of each detected pattern is to be tracked. For example, if you specify the item 'PRICE>100EUR' as a `TrackedItem`, you will be shown for every detected association how many of the data records or data groups in which the association occurs have a price of more than 100 EUR.
- **<AssociationsResultSpec .../>**: defines various settings for exporting association models. The element has the following (optional) attributes:
  - **format**: output format of the model ("FLAT_FILE", "FLAT_FILE_NO_-HEADER", "PMML" or "JDBC_TABLE")
  - **colSeparator**: column separator character to be used in the output model (only required in the output formats "FLAT_FILE" and "FLAT_FILE_NO_-HEADER"). Default value is <TAB>.
  - **writeToStdOut**: if this parameter is set to 'true', the model will be written both to the standard output console (`stdOut`) and to the specified output file.
  - **description**: textual description of the association model.
  - **writeChiSqrConf**: 'true' or 'false'. Indicates whether the χ^2 confidence of each association is to be written into the model. Per default, chi-square confidences are written if and only if a `minChiSqrConf` filter greater than 0.0 has been set.
  - **writePurities**: 'true' or 'false'. Indicates whether the purity of each associations is to be written into the model output. Default is 'true'.
  - **writeWeight**: 'true' or 'false'. Indicates whether the weight (price, cost) of each association is to be written into the model output. Per default, weight is written if and only if a weight/price field has been specified on the training data.
  - **writeConfidences**: 'true' or 'false'. Indicates whether the model output should contain the confidences of all possible if-then rules which can be formed from a given association within the model by taking one of the association's items as 'then' side and all other items as the 'if' side of the rule.
  - **writeItemSupports**: 'true' or 'false'. Indicates whether the occurrence frequencies (absolute supports) of each single item within each association are to be written into the model output. Default is 'true'.

  - **writeSupportGroups**: 'true' or 'false'. Indicates whether up to 3 sample data records or data groups out of the support of each association are to be written into the model output. Default is 'false'.
  - **itemMode**: "SINGLE" or "COMBINED". Indicates whether the names of the single items which form a association are to be written into separate columns of the model output or into one single column containing all item names. This setting is irrelevant for the output format 'PMML'. Default value is "SINGLE".

- **<ResultDataLocator .../>**: defines name, access path and data format of the file or database table into which the result of the associations analysis is to be exported. The internal structure of this element has been described in subsection <DataLocator>.

<SequencesTrainTask>

<SequencesTrainTask> defines the task to perform a sequential patterns analysis and to generate a collection of sequential patterns on the data described in the <InputData> section. The result can be returned in the form of a PMML <SequenceModel> or in tabular form as a flat file.

<SequencesTrainTask> can contain the same attributes and subelements as <AssociationsTrainTask>. However, between formally identical attributes and subelements in <AssociationsTrainTask> and <SequencesTrainTask> there is the semantic difference that 'support' means something different in sequences compared to associations. In associations, support refers to the number of data records or data groups (transactions) in which a pattern occurs. In sequences, support refers to the number of 'entities' for which transaction data have been collected. For example, in market basket analysis, the support of an association is the number of sales slips (transactions) in which a combination of articles occurs, whereas the support of a sequence is the number of customers (entities) for whom a certain time-ordered purchasing pattern applies.

A sequential patterns analysis can only be performed if the <InputData> section of the task specification defines an 'ENTITY' data field and an 'ORDER' data field.

<SequencesTrainTask> can contain the following additional subelements which can not occur in <AssociationsTrainTask> or which have a different meaning there:

- **<NbItems min="..." max="..."/>**: lower and upper limit for the number of single bits of information ('items') in the sequential patterns to be detected.
- **<PatternLength min="..." max="..."/>**: lower and upper limits for the number of 'item sets' (events) in the sequences to be detected. Each 'item set' consists of one or more atomic bits of information ('items') which occur at the same time. Hence, <PatternLength> is the number of time steps in the sequence plus 1.
- **<ItemsetLength min="..." max="..."/>**: lower and upper limits for the number of atomic bits of information ('items') which can be contained in a single event ('item set') which can appear in the sequences to be detected.

- **<SequencesResultSpec ... />**: has the same function as the element <AssociationsResultSpec> in <AssociationsResultSpec> and contains exactly the same attributes.
- **<ResultDataLocator ... />**: defines name, access path and data format of the file or database table into which the result of the sequences analysis is to be exported. The internal structure of this element has been described in subsection <DataLocator>.<

<RegressionTrainTask>

<RegressionTrainTask> defines the task to perform a regression analysis and to generate a regression model on the data described in the <InputData> section. The result can be returned in the form of a PMML <RegressionModel> or in tabular form as a flat file.

<RegressionTrainTask> can contain the following optional attributes:

- **maxNbRegressors**: maximum number of regressor variables, that means data fields which appear on the left hand side of the regression equation to be created. If there are more active data fields, a selection will be performed based on the fields' importance (regression coefficient strength) and on field-field correlations.
- **missingValueReplace**: specifies how missing values in regressor fields are to be handled. Possible values are:
  "ZERO" (default): replaces missing values by 0.
  "MEAN": replaces missing values by the field's mean value.
  "SKIP_RECORD": ignores every data record in which at least one active regressor field has no valid value.
- **withConstantOffset**: specifies whether the regression equation can contain a constant term (offset). Default is 'true'.
- **createResidualField**: if this parameter is 'true', a new data field named 'RESIDUAL' will be created in the training data. The new data field contains the model's prediction error for each data record, that is the residual (actual target field value minus predicted target field value). Default value is 'false'.

<RegressionTrainTask> can contain the following subelements:

- **<RegressionResultSpec ... />**: defines various settings for exporting regression models. The element has the following (optional) attributes:
  - **format**: output format of the model ("FLAT_FILE", "FLAT_FILE_NO_HEADER", "PMML" or "JDBC_TABLE"). Default value is "FLAT_FILE"
  - **colSeparator**: column separator character to be used in the output model (only required in the output formats "FLAT_FILE" and "FLAT_FILE_NO_HEADER"). Default value is <TAB>.

  - **writeToStdOut**: if this parameter is set to 'true', the model will be written both to the standard output console (`stdOut`) and to the specified output file.
  - **description**: textual description of the regression model.
  - **writePredictedError**: 'true' or 'false'. Specifies whether the mean prediction accuracy (root mean squared error) on the training data is to be written into the model. Default is 'true'.

- **<ResultDataLocator ... />**: defines name, access path and data format of the file or database table into which the result of the regression analysis is to be exported. The internal structure of this element has been described in subsection `<DataLocator>`./LI>

<SOMTrainTask>

<SOMTrainTask> defines the training of a self organizing map (SOM model) - that means a two-dimensional grid of neurons - on the data described in the `<InputData>` section.SOM models can be used for cluster analysis and for prediction of unknown data field values. The resulting SOM model can be returned in the form of a PMML `<ClusteringModel>` or in a proprietary binary format.</en_US>

`<SOMTrainTask>` can contain the following optional attributes:

- **nbVerificationRuns**: number of control models, which are built with the same parameter settings as the main model but with different random initializations of the neuron weights. The comparison between the main model and the control model(s) indicates whether the main model is well converged.
- **maxNbIterations**: maximum number of training iterations of the SOM net
- **targetWeight**: multiplication factor for the relative weight of the 'target' data field compared to the other active data fields. Default value is 1.0. Setting the parameter to values grater than 1 results in SOM models in which the SOM card for the target field shows a clearer distinction between low-target-value regions and high-target-value regions.
- **nbNeuronsX**: number of SOM neurons in x direction
- **nbNeuronsY**: number of SOM neurons in y direction.
- **createResidualField**: if this parameter is 'true', a new data field named 'RESIDUAL' will be created in the training data. The new data field contains the model's prediction error for each data record, that is the residual (actual target field value minus predicted target field value). Default value is 'false'.

`<SOMTrainTask>` can contain the following subelements:

- **<SOMResultSpec ... />**: defines various settings for exporting SOM models. The element has the following (optional) attributes:
  - **format**: output format of the model ("BINARY" or "PMML")

  - **writeToStdOut**: if this parameter is set to 'true', the model will be written both to the standard output console (`stdOut`) and to the specified output file.
  - **description**: textual description of the SOM model.
  - **writePredictedError**: 'true' or 'false'. Specifies whether the mean prediction accuracy (root mean squared error) on the training data is to be written into the model. Default is 'true'.Wenn dieses Attribut auf `"true"` gesetzt ist, wird die mittlere Vorhersagegüte (die Quadratwurzel des mittleren quadratischen Fehlers) des SOM-Modells bei der Vorhersage der Zielfeldwerte auf den Trainingsdaten in das exportierte Modell geschrieben. Voreinstellung ist `"true"`.

- **<ResultDataLocator .../>**: defines name, access path and data format of the file or database table into which the result of the SOM training is to be exported. The internal structure of this element has been described in subsection `<DataLocator>`.<

4.2 The command line processor sacl

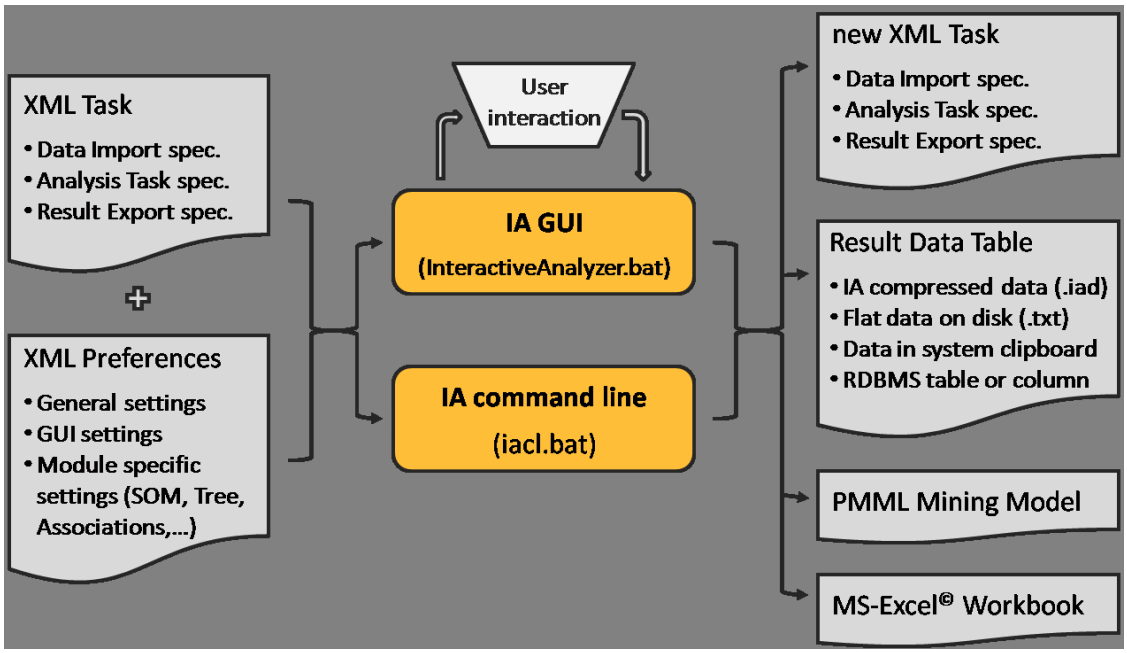
4.2.1 The command line processor sacl

Based on an XML interface, Synop Analyzer can be used as an analysis kernel within automated workflows or batch processes, or as a plugin component embedded into third-party software. To that purpose, the Synop Analyzer command line processor (`sacl.bat`) can be used. It processes an analysis task - submitted in the form of an XML document - without user interaction.

`sacl.bat`can take 1 or 2 command line parameters:

- An analysis task in the form of an XML document which can be validated against the XML schema `http://www.synop-systems.com/xml/InteractiveAnalyzer-Task.xsd`,
- The name of the XML file which contains the preference settings to be used. This file must validate against the XML schema `http://www.synop-systems.com/xml/-InteractiveAnalyzerPreferences.xsd`.

The result of calling the command line processor can either be a transformed version of the input data or an analysis result in the form of a report (HTML or PDF), a spreadsheet (.xlsx) with tabular and graphical information, a data table or a data mining model. The following picture shows this schematically:



4.2.2 XML analysis task specifications

XML tasks according to the XML schema http://www.synop-systems.com/xml/InteractiveAnalyzerTask.xsd are described in detail in The Synop Analyzer XML Application Programming Interface.

4.2.3 Examples

A simple task, which reads the flat file kunden.txt from the subdirectory doc/sample_data of the Synop Analyzer installation directory, creates a value distribution statistics and chart for each data field and writes the result to a spreadsheet called kunden_stat.xlsx, is given below:

```
<?xml version="1.0" ?>
<InteractiveAnalyzerTask>
   <InputData>
      <InputDataLocator usage="DATA_SOURCE" type="FLAT_FILE"
         name="doc/sample_data/kunden.txt"/>
   </InputData>
   <UnivariateExplorationTask nbChartsPerRow="3">
      <ResultDataLocator usage="IA_REPORT" type="OOXML_SPREADSHEET"
         name="doc/sample_data/kunden_stat.xlsx"/>
   </UnivariateExplorationTask>
</InteractiveAnalyzerTask>
```

If you store this task as kunden_task1.xml and create a batch file kunden_task1.bat containing the line of text `sacl kunden_task1.xml`, then you can call kunden_task1.bat from command line, from a shell script or a scheduled service or workflow or by mouse click in order to execute the specified task without any further user interaction.

You could also submit the XML task directly as a textual string when calling sacl. In this case, however, you have to 'quote' the task by enclosing it into double quotes. The existing double quotes within the string have to be masked by backslashes (\) in this case. The call would then look like this:

```
c:> SynopAnalyzer.bat "<?xml version=\"1.0\" ?><InteractiveAnalyzerTask>
<InputData><InputDataLocator usage=\"DATA_SOURCE\" type=\"FLAT_FILE\"
name=\"doc/sample_data/kunden.txt\"/></InputData>
<UnivariateExplorationTask nbChartsPerRow=\"3\">
<ResultDataLocator usage=\"IA_REPORT\" type=\"OOXML_SPREADSHEET\"
name=\"doc/sample_data/kunden_stat.xlsx\"/></UnivariateExplorationTask> </Interactive-
AnalyzerTask>"
```

4.3 Task automization and workflows

(editing in progress)

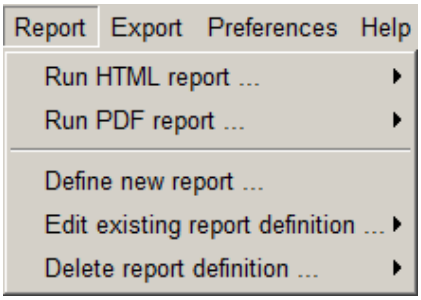
4.4 Defining and Running Reports

4.4.1 Concept

The reporting functions of Synop Analyzer are started from the main menu item **Report**. The purpose of the reporting module is to generate printer-read high-quality PDF reports or web-ready HTML reports for online publishing from one or more data analyses which have been executed in Synop Analyzer. By means of these reports, you can communicate %IA; data analysis results or store them in a revision-safe format for regulatory or auditing purposes. You can optically adapt the reports to your company's guidelines and corporate design templates by referring to external CSS stylesheets

Generating reports involves two steps, which can be performed by different people: first you define a report template, then (and maybe repeatedly at later times) you create the HTML or PDF report itself by 'running' a predefined report template on the current data available at that time. Report templates can be run from the graphical workbench but also in an automatic mode using the Synop Analyzer command line processor.

The picture below shows the different features of the menu item **Report**:



The three menu items below the horizontal separator line serve to create, edit or delete report templates. The two menu items above the separator create a HTML or PDF report out of a report template, taking the currently available in-memory data within Synop Analyzer to fill the place holders in the template with up-to-date charts, tables and figures.

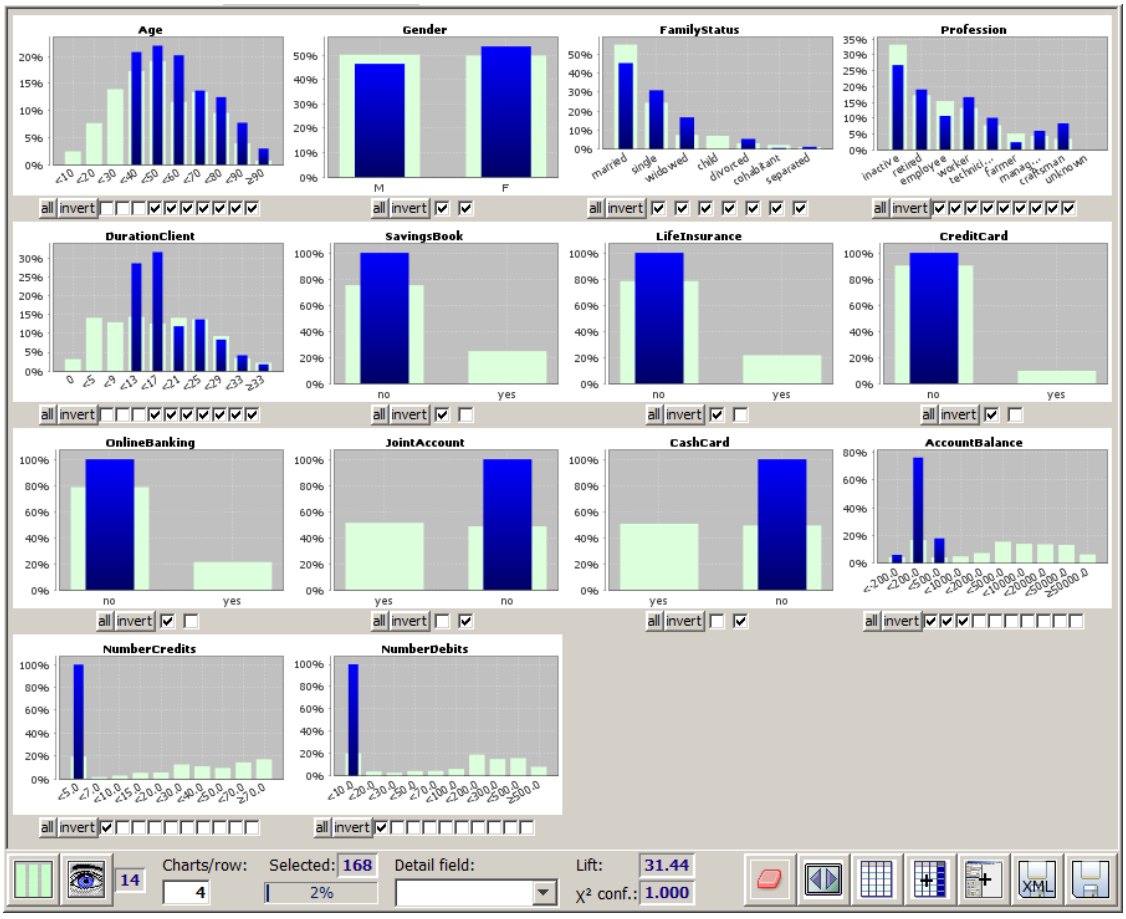
4.4.2 A sample use case

The whole workflow from defining and refining a report template, linking an external CSS stylesheet up to creating the final report shall be demonstrated at hand of a simple example. In the example, we assume that the sample data `doc/sample_data/customers.txt` are the customer master data of the Newtown affiliation of Frist Profit Bank, and that these data are to be monitored and quality assured once a year. The goal of the monitoring is to detect presumable data errors and to identify inactive customers. The results are to be documented in an 'revision safe' manner in the form of PDF documents which will be stored in a repository.Furthermore, we assume that First Profit Bank has a corporate

design template for its official online and printed documents, a CSS stylesheet named `doc/sample_data/report_stylesheet.css`. This stylesheet is to be used for formatting our report.

The desired analysis tasks of our example can be performed by loading the data `doc/sample_data/customers.txt` into Synop Analyzer and by opening three analysis tabs on them:

- An analysis tab of type 'Statistics and Distributions' gives us a first overview over the data and reveals possibly missing data or strange attribute values.
- An analysis tab of type 'Deviations and Inconsistencies' is used to detect more subtle data errors such as customers whose stored demographic data do not match with their customer behavior such as accounting activity or fortune balances.
- An analysis tab of type 'Multivariate Exploration' detects the customers which have been inactive over a long period of time.

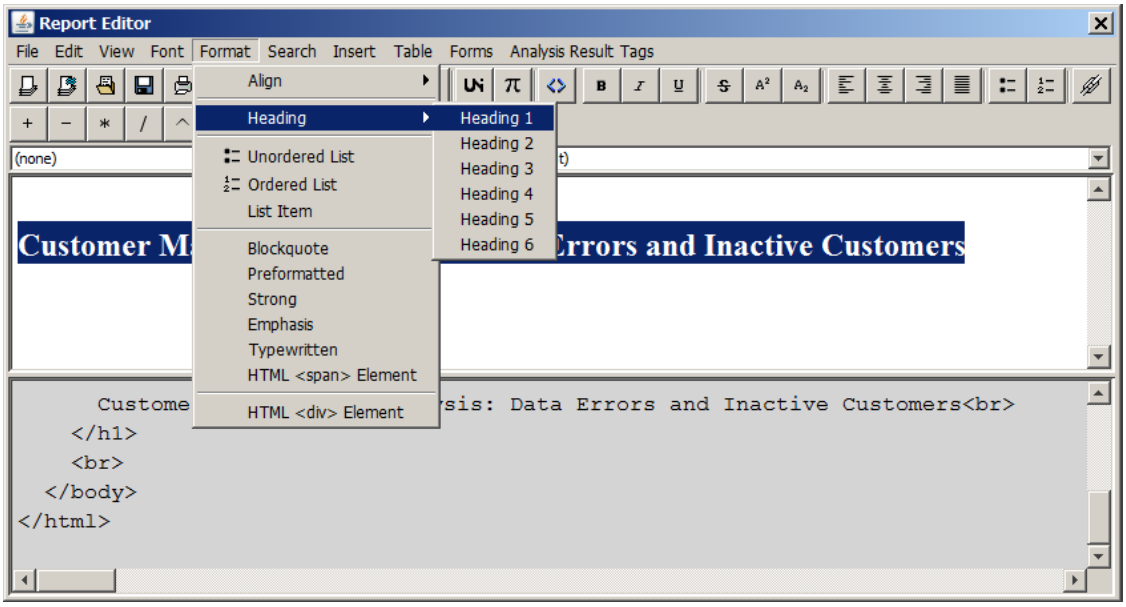


You can load all these analysis tasks and tabs via the main menu item **Project → Open** by selecting the project file `doc/sample_data/project_customer_masterdata_monitoring.xml`.

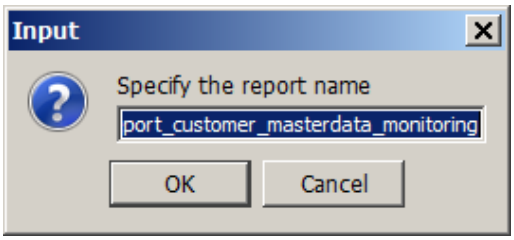
4.4.3 The Visual Report Designer

Via **Report → Define new report** we now create the desired report template. The menu item opens a visual report editor window. In the upper part of the window, you can type and format the report's headings and texts just like in a typical word processing program. The editor creates HTML source code which canbe introspected (and modified) in the lower part of the window with gray background color.

In the screenshot shown below, we have typed in the desired head line of the report; then we have formatted the head line using the editor menu item **Format → Heading 1**.



Once a report template has been created, you can leave the report editor by simply closing the editor window. A pop-up dialog will appear which asks for a name for the newly created report template:



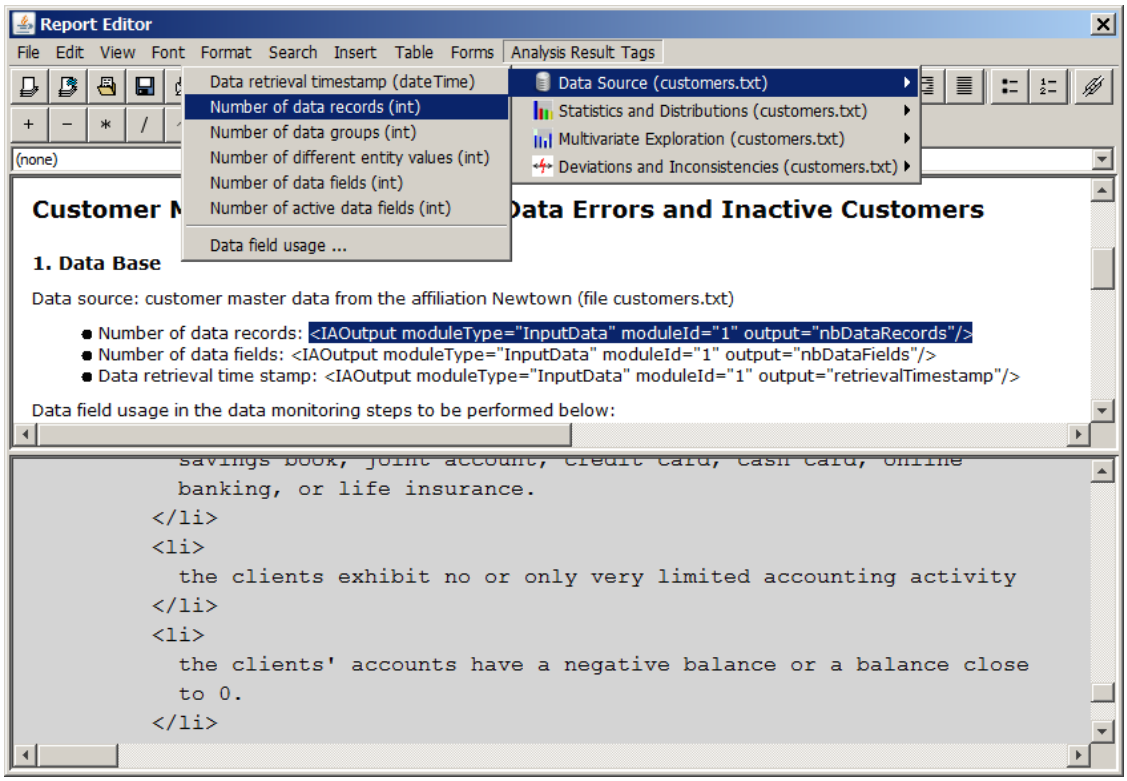
Attention: When you leave the report editor and specify a name for the new report template, the template is stored in-memory as a part of the currently opened Synop Analyzer project. You have to save the project via **Project → Save** before you leave the Synop Analyzer workbench or before you close the current project by closing all its input data tabs, otherwise your editings and your new report template will be lost.

4.4.4 Linking Synop Analyzer analysis results

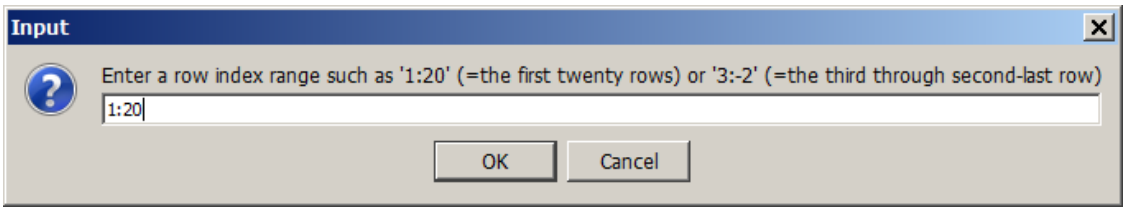
The report editor offers several functions beyond the scope of a word processing program. In particular, it has the ability of placing links to charts, tables and figures which appear in the currently opened input data tabs and data analysis tabs within the Synop Analyzer GUI. The link is not a hard link which just copies the current content of the link destination. Instead, this copying action is only performed when the report template is 'executed' and a final HTML or PDF report is created from it. That means, when you 'run' the report template in the future, the resulting report will always reflect the must up-to-date data. The editor's menu item **Analysis Result Tabs** serve to place such a 'soft' link to an analysis result into the template. This menu item contains several groups of sub-items: one group for each currently opened analysis or input data tab.

Attention: Some of Synop Analyzer's analysis modules, for example the module 'Deviations and Inconsistencies' which we have used in our example, do not automatically create their analysis results when the panel is opened. Instead, a button such as **Start the training** has to be pressed and a possibly long-running background process creates the analysis result. When you want to link a result from such an analysis tab into a report template, you have to assure that the background process has been started and completed before you open the report editor. Alternatively, you can activate the checkbox 'Automatically execute all asynchronous tasks' in **Project → Project Settings**.

The screenshot displayed below shows all possible results which can be embedded in a report from an input data tab:

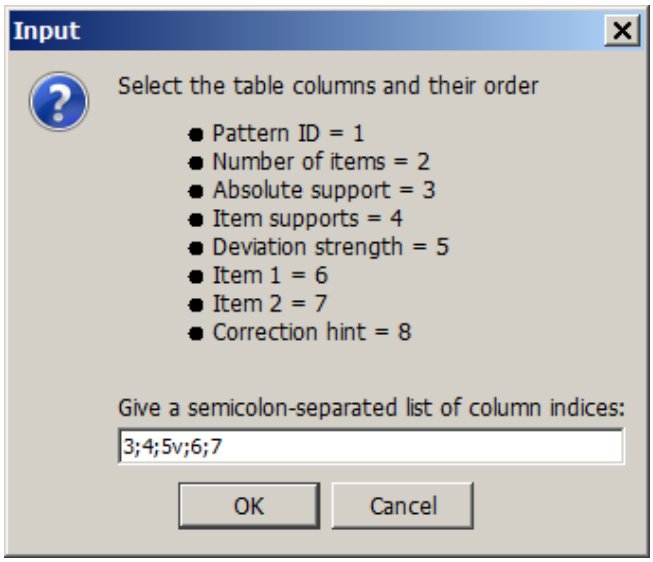


The embedded links appear within the report template in the form of tags starting with code><IAOutput .../></code>.Some embedded links require further specifications. The editor asks for these additional settings by means of pop-up dialogs. For example, when linking to a table, the question pops up which table rows and which table columns are to be shown, and in which order. In the screenshot shown below, we have specified that not more than the first 20 rows of the table are to be shown in the report.

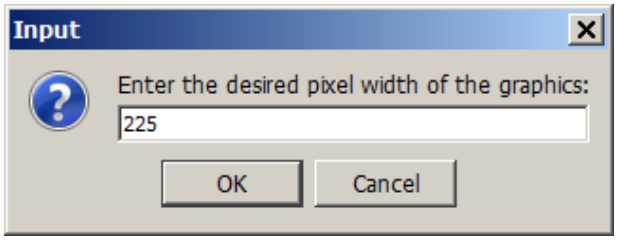


Another pop-up dialog asks for a column selection and whether the table rows are to be sorted by ascending or descending values of a certain table column.

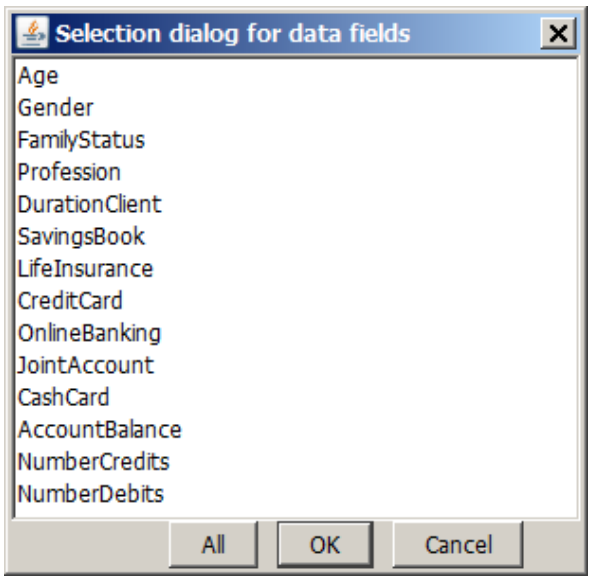
In the dialog shown below, we have specified that only the table columns 3 to 7 (in this order) are to be displayed in the report. The table rows are to be sorted by descending values of column 5. This is specified by the character 'v'; ascending ordering would have been specified by the 'circumflex' character, that means the letter 'v' rotated by 180 degrees.



For graphical results you have to specify the desired width (in pixels):

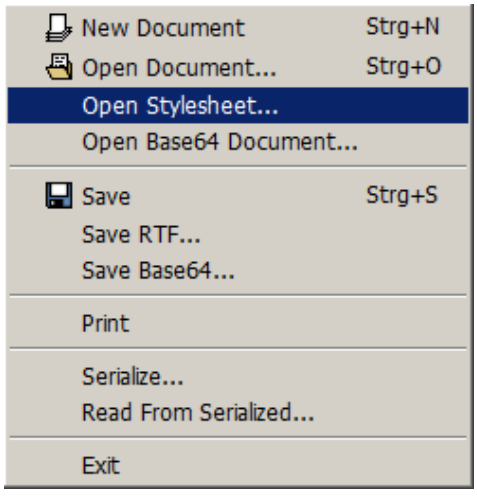


For series of graphical outputs one can specify the subset of the series to be displayed in the report:



4.4.5 Using Stylesheets

Using predefined layouts and stylesheets in a report template is possible by embedding an external CSS stylesheet into the report template. This is done in the report editor via the menu item **File → Open Stylesheet**

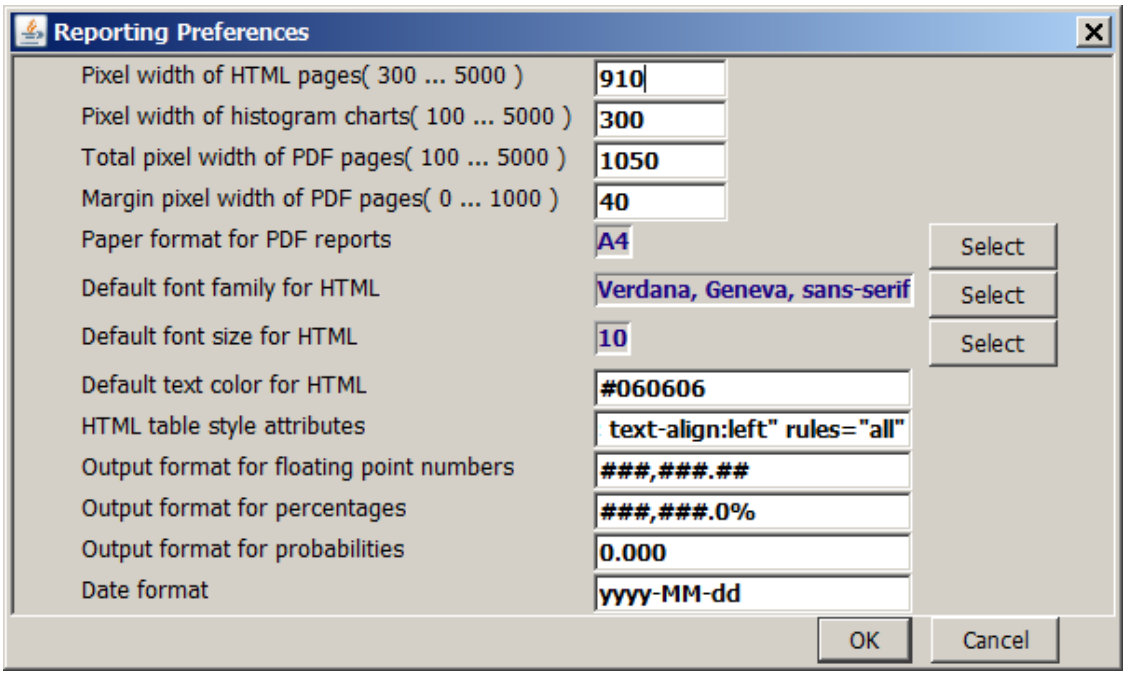


In our example, we use the predefined stylesheet `doc/sample_data/report_stylesheet.css`.**Achtung** most stylesheets divide the available space on each screen page in different areas called 'divisions' which are specified using the HTML tag `<DIV .../>`. the stylesheet `doc/sample_data/report_stylesheet.css`, for example, expects that the main part (`<BODY>`) of the HTML document is divided in the following divisions:

```
<DIV id="container">
  <DIV id="header"></DIV>
  <DIV id="main_part">
    ...(the report content must appear here)...
  </DIV>
</DIV>
```

These <DIV>-Elements can be added to the report template using the menu item **Format → HTML div**. You can also add the <DIV> tags to the report template by editing the XML project file which contains the report template using an arbitrary XML editor.

There are some preference settings for reports. They can be modified via the main menu item **Preferences → Reporting Preferences** and influence the layout of the resulting HTML and PDF reports. When creating PDF reports, you should make sure that the pixel width of the pages of your PDF reports are about 10% to 20% larger than the pixel width of the 'usable' range defined in the used CSS stylesheet.

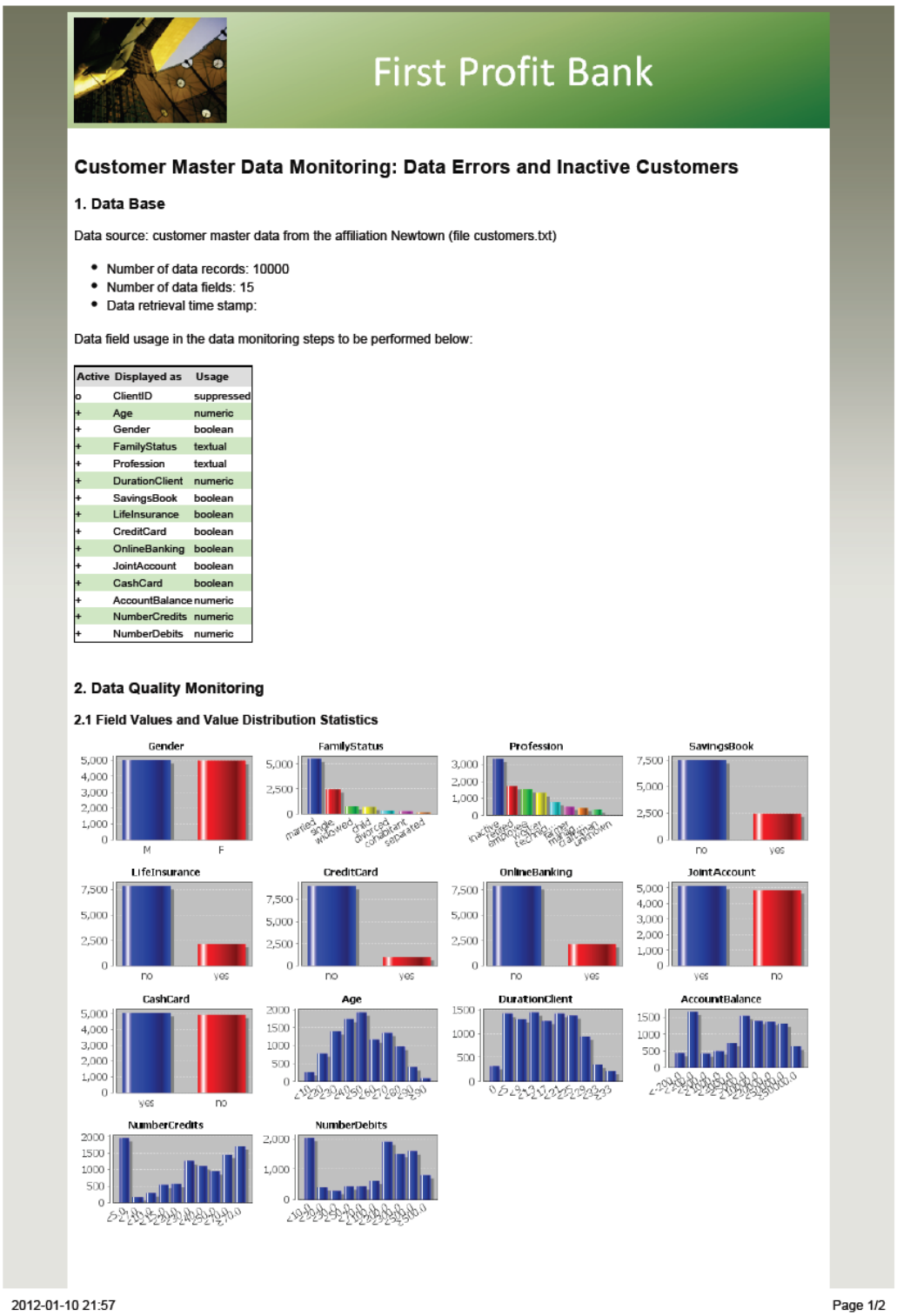


If the text sizes of the chart labels in the graphs within the generated reports are too small or too large, you should edit the preference setting 'default chart label font' in the preferences dialog **Preferences → GUI Preferences**.

4.4.6 Creating HTML or PDF Reports

Once a report template has been defined within an Synop Analyzer project, it can be executed ('run') via the main menu item **Report**. 'Running' a report template means replacing the analysis result links in the template by the most up-to date analysis results available in the Synop Analyzer GUI and exporting the resulting report to a HTML file

(plus several PNG picture files) or to a PDF file. The PDF report generated for our example looks like this:



First Profit Bank

Customer Master Data Monitoring: Data Errors and Inactive Customers

1. Data Base

Data source: customer master data from the affiliation Newtown (file customers.txt)

- Number of data records: 10000
- Number of data fields: 15
- Data retrieval time stamp:

Data field usage in the data monitoring steps to be performed below:

Active	Displayed as	Usage
o	ClientID	suppressed
+	Age	numeric
+	Gender	boolean
+	FamilyStatus	textual
+	Profession	textual
+	DurationClient	numeric
+	SavingsBook	boolean
+	LifeInsurance	boolean
+	CreditCard	boolean
+	OnlineBanking	boolean
+	JointAccount	boolean
+	CashCard	boolean
+	AccountBalance	numeric
+	NumberCredits	numeric
+	NumberDebits	numeric

2. Data Quality Monitoring

2.1 Field Values and Value Distribution Statistics

5

Step-by-step Tutorials

In this part of the user's guide we present a collection of step-by-step tutorials which demonstrate the handling and features of the Synop Analyzer modules 'by example'. All use cases have been documented and described in detail with many screenshots, and all use cases are based on one of the sample data sources which come with Synop Analyzer. Therefore, you can reproduce each step one to one on your computer.

Interactive Customer Intelligence: Based on a Multivariate Exploration (i.e. an iteractive multidimensional ad-hoc drill-down) into a customer master data table, we detect sales potentials and select a suitable target group for a sales campaign. The tutorial uses the sample data `kunden.txt`.

5.1 Tutorial: Customer Intelligence

5.1.1 Business Case

Understanding the customers and their needs is essential for every enterprise in today's economic environment, which is largely characterized by supply surpluses and selective and critical customers who are well aware of the fact that they can choose among many different products and vendors.

With its intuitive and highly scalable multidimensional data exploration capabilities, Synop Analyzer provides a powerful tool for a data-driven approach to understanding customer behavior, and for deriving sales and marketing strategies from these insights. In this tutorial, we demonstrate this unique approach using bank customer master data with a view to analyzing those customers with large amounts of money on their current accounts. The questions to be answered are as follows:

- Can customers with a high average balance on their current account be clustered into several homogeneous groups of people with similar demographic or buying attributes? Which attributes describe these clusters?
- Can we derive sales and marketing strategies, up-selling actions or other sales or marketing campaigns from these findings?

Other application scenarios for this type of interactive data exploration comprise:

- Marketing campaign planning: select suitable target groups and the best matching contact channel for each group within a marketing campaign for a specific product or service offering.
- Sales planning and marketing strategies.
- Sales controlling and success tracking: which sales unit or sales representative performed better/worse than their peer groups and why? In what respect did the successful sales units differ from the less successful sales units?
- Sales force education, training and support: provide each sales representative with individual hints and success strategies tailored towards the specific characteristics of the sales representative's client group, region, product portfolio, time of the year etc.

5.1.2 Advantages of the Synop Analyzer approach to Customer Intelligence

Compared to other methods and tools for customer data analytics, Synop Analyzer offers the following advantages:

- **Ease of use:** Unlike many other interactive 'drill-down' tools, Synop Analyzer does not require elaborate data preprocessing or data modeling (e.g. the setup of a cube model and a cubing engine) prior to starting the exploration itself. Just connect to a raw data table or view, or load an Excel sheet or flat file and start the exploration.
- **Speed:** Typically, you will have your exploration results only minutes after connecting to a table or file. This quick 'time to result' approach can be achieved because Synop Analyzer automatically performs the necessary data attribute selections and data preparations (e.g. value discretizations into suitable ranges or intervals), and because the single drill-down and data selection steps are truly 'interactive' in the sense that they return their results within fractions of a second, even on the largest customer data set.
- Unique combination of **interactive data exploration and data mining:** together with the interactive data exploration capabilities demonstrated in this tutorial, Synop Analyzer offers a set of powerful data mining features whose capabilities complement the features demonstrated in this tutorial. Separate tutorials are available for these features.
- **Performance:** due to its data compression scheme, its in-memory data handling and its powerful analytics algorithms, Synop Analyzer is able to explore data tables of 10 million customers or more on a simple PC or notebook with 'real-time' response times of less than a second.
- **Scalability:** Synop Analyzer's analytic engine supports multithreading, multi-core CPUs and multi-CPU servers with almost perfectly linear speedup. That means, if you double the number of concurrent users, doubling the number of available CPU cores and the available RAM on your analytics server keeps the engine's response times unchanged.
- **Seamless integration** into the existing IT and business analytics infrastructure: Synop Analyzer interacts with databases, reporting systems and other enterprise applications via standard interfaces such as JDBC, web services (SOA), or XML. You can define and deploy Synop Analyzer standard processes in the form of automatically running batch jobs, database stored procedures or system services.
- **Competitive TCO** (total cost of ownership) due to little demand for hardware, software and administrative resources and flexible pricing. This makes Synop Analyzer suitable for virtually all companies, regardless of IT department sizes and budgets.

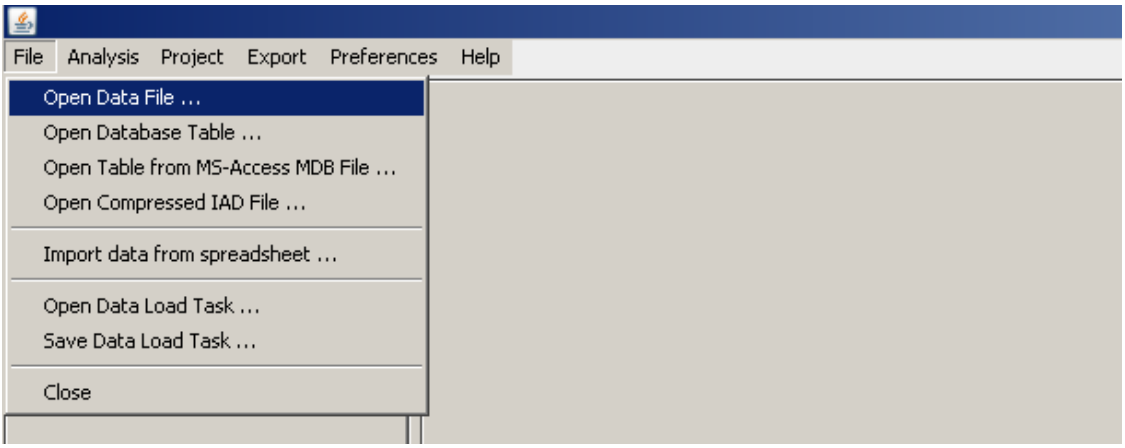
5.1.3 Sample Data used in this Tutorial

In this tutorial, we analyze a master data file containing customer data of 10,000 bank customers. The data records are available in the form of the `<TAB>` separated flat file `doc/sample_data/customers.txt` which contains 15 attributes such as the customers' age, profession, family status, customer history, assets and the banking services they are using.

5.1.4 Step 1: Loading the Data

In this section, we use the Synop Analyzer module Input Data to load a flat text file into Synop Analyzer. We start the Synop Analyzer workbench (double-click on the executable batch file `SynopAnalyzer`)The main panel of the Synop Analyzer graphical workbench opens up, showing a bipartite empty canvas. The left column will later display some basic properties of all data sources which have been opened in Synop Analyzer. In the right part of the canvas you can run various data analysis modules.

Using the main menu item **File** we can select a flat text file to be opened:



Clicking on **File** → **Open Data File** opens up a file chooser dialog in which we select the input data file `doc/sample_data/customers.txt`.

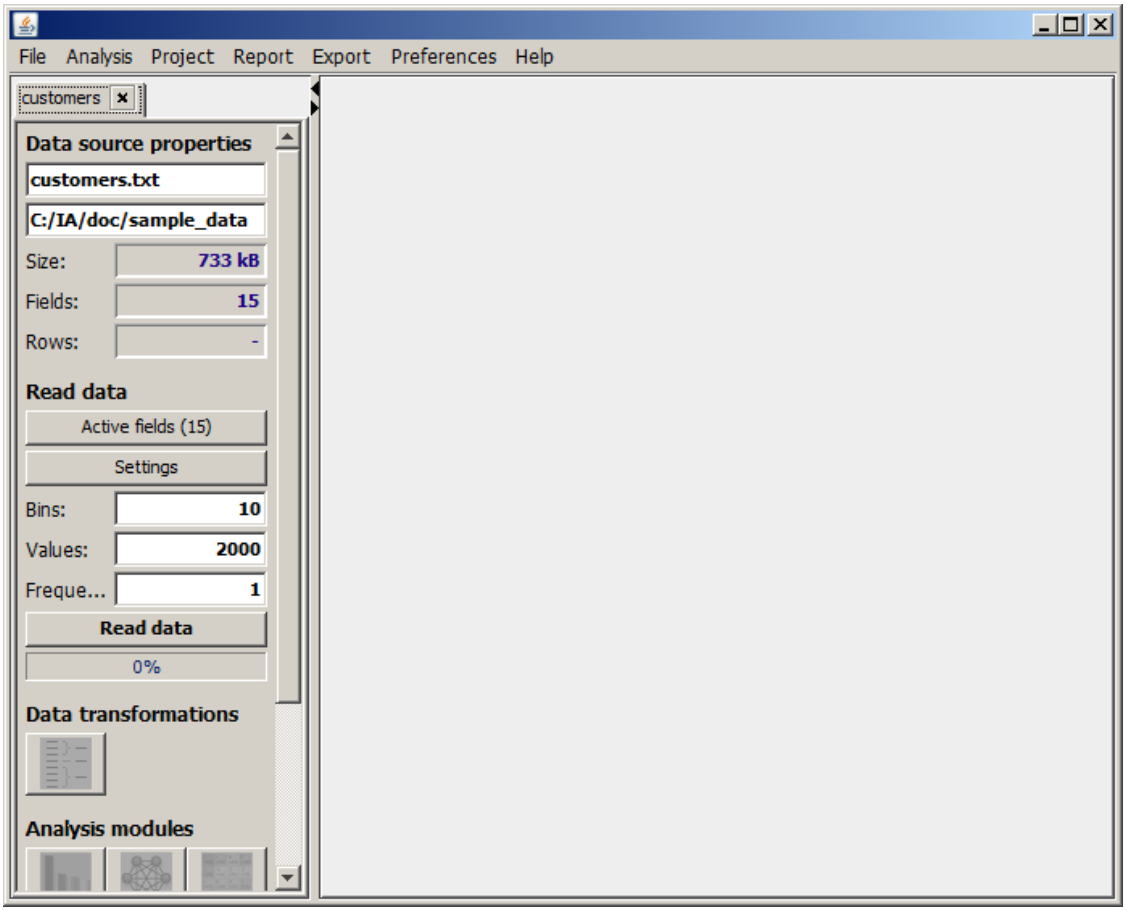
Now the left part of the screen displays some basic properties of the data. In the following, we will call that part of the screen the input data panel. At the beginning of each data exploration with Synop Analyzer, the data has to be loaded into a binary compressed representation which resides in the computer's RAM. This loading process is started by pressing the **Start** button in the input data panel. In addition, the panel provides a couple of input fields and buttons for manually adjusting the data import process. For now, we want to load the input data using default import settings, therefore we directly press the **Start** button.

Note: every data specification and analysis module of Synop Analyzer has a context-sensitive help system in the form of a 'mouse-over' function which opens up explaining texts for a button, input field or output element whenever you place the mouse pointer on a label, field or button.

Note: you find more explanations on the advanced parameters for the data loading in the module description of the Input Data Panel, for example how to modify the number of value ranges or the range boundaries shown in the histograms for the numeric data fields.

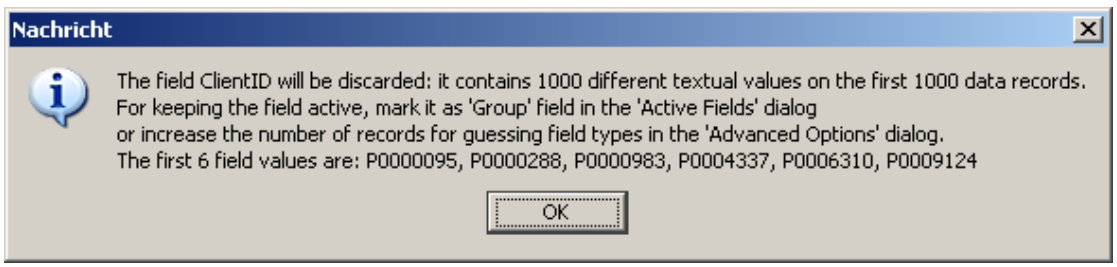
Note: by opening the pop-up dialog **Show advanced options** in the input data panel and by activating the checkbox **Create persistent data file** you can create a permanent

version of the compressed data on disk. The compressed file has only about 8% to 10% of the initial data file size. This data file can be re-read in later analysis sessions, which for large files is much faster than reading and compressing the corresponding flat file each time.



While the blue progress bar proceeds from 0 to 100%, the data is read, compressed and stored in memory. On large data sets, on a typical PC with one CPU, about 1 GB of data can be read and compressed per minute.

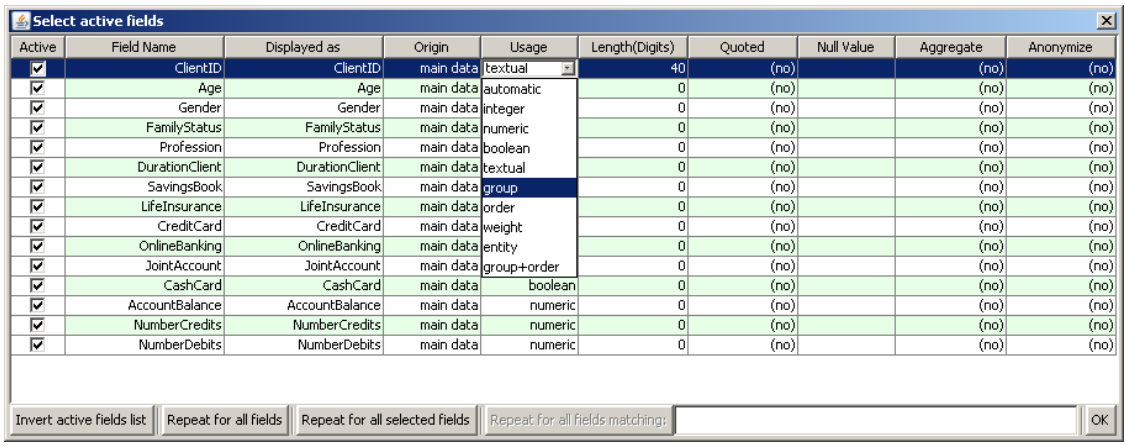
When the data reading is finished, the buttons in the lower part of the input data panel change their appearance, indicating that the data are now ready to be used. But In our case we first get a warning message:


Nachricht
The field ClientID will be discarded: it contains 1000 different textual values on the first 1000 data records.
For keeping the field active, mark it as 'Group' field in the 'Active Fields' dialog
or increase the number of records for guessing field types in the 'Advanced Options' dialog.
The first 6 field values are: P0000095, P0000288, P0000983, P0004337, P0006310, P0009124
OK

The message tells us that the column `ClientID` is a key-like field which contains a unique value in each data row. Those fields are not suitable for creating value distribution statistics or for using them as selection criterion within interactive data analysis steps. Therefore, the data field is being deactivated by default.

Even though we agree that we don't want to see ClientIDs in statistics or multivariate data explorations, we would like to maintain the field in the imported data because the values serve as unambiguous identifiers (keys) for the data sets: Whenever we have selected an interesting set of data records, we need their `ClientID`'s in order to unambiguously identify the selection's data records.

Therefore we follow the advice of the warning message and open the pop-up dialog **Select active fields**. In the leftmost column (**Active**), we re-activate the field `ClientID`. In the column **Usage** we define the field to be the group field:



Now we close the pop-up window with the **OK** button and reload the data by pressing **Start**. This time, the data reading succeeds without warning messages. Once the data are available in memory, the buttons in the lower part of the input data panel on the left side of the screen become usable. We first press the button **Statistics and Distributions** in order to get a quick overview of the available data fields and the data quality.

5.1.5 Step 2: Obtaining a First Overview

The main purpose of the 'Statistics and Distributions' panel is to gain a first overview on the kind of information contained in the data. Furthermore, obvious data quality issues become visible, such as fields with many missing or invalid values, fields with erroneous values (e.g. negative age, profession="xxx", etc.). A more sophisticated data quality checking can be performed using the module Deviations and Inconsistencies which is described in a separate tutorial.

The panel 'Statistics and Distributions' consists of three parts separated by horizontal bars. By mouse-drawing these bars, or by clicking on the arrow symbols on the left end of the bars, you can change their size and minimize or maximize each of the three parts.

The first part shows an overview statistics on the active numeric data fields: the original and the displayed field name, the number of rows with missing or invalid (non-numeric) content, the number of different values, and the basic statistical distribution measures such as mean, median, minimum, maximum, standard deviation, etc.

The second part shows an overview statistics on the active textual and Boolean fields: the field name, the number of rows with missing content, the number of different values, the most frequent value with its frequency and the second most frequent value with its frequency.

The third part displays a graphical representation of each field's value distribution in the form of one histogram chart per data field.

For the numeric fields, Synop Analyzer has automatically chosen suitable discretizations into the number of bins that has been specified in the field **#bins (numeric field)** on the input data panel. Depending on a field's actual value distribution statistics, Synop Analyzer either chooses a binning into equidistant intervals, or a logarithmic binning. In the data used here, the field `Age`, which has a value distribution close to a Gaussian 'normal' curve, has been discretized into equidistant intervals. The field `AccountBalance`, on the other hand, has been discretized logarithmically. The software has automatically detected that this field's value distribution is not suitable for equidistant binning because it has its center between -200 and 1000 but also a significant 'fat tail' at much higher values of more than 10000 or even 50000.

Statistics and Distributions
Numeric/Group data field | Invalid ... | Diff. val... | Minimum | Maximum | Second sma... | Second largest | Mean | Median | Std. deviation | Skewness | Excess
Age | 0 | 100 | -1.0 | 104.0 | 0.0 | 97.0 | 45.3903 | 45.0 | 20.18693 | 0.1332122 | -0.7239484
DurationClient | 0 | 53 | 0.0 | 77.0 | 1.0 | 51.0 | 14.4973 | 15.0 | 9.095768 | 0.30050248 | -0.37858394
AccountBalance | 0 | 6522 | -2388000.0 | 2112000.0 | -227000.0 | 463300.0 | 12964.228 | 3500.0 | 41946.773 | -4.873503 | 1706.083
NumberCredits | 0 | 269 | 0.0 | 2290.0 | 1.0 | 799.0 | 40.6366 | 35.0 | 51.74998 | 11.7661915 | 387.1006
NumberDebits | 0 | 838 | 0.0 | 1412.0 | 1.0 | 1375.0 | 189.7215 | 150.0 | 194.92775 | 1.3645356 | 2.2274115
Textual data field | Invalid or NULL | Diff. values | Other values | Most freq. value | Highest freq. | Second most freq. ... | 2nd highest freq.
Gender | 0 | 2 | 0 | M | 5019 | F | 4981
FamilyStatus | 0 | 7 | 0 | married | 5494 | single | 2440
Profession | 0 | 9 | 0 | inactive | 3328 | retired | 1727
SavingsBook | 0 | 2 | 0 | no | 7526 | yes | 2474
LifeInsurance | 0 | 2 | 0 | no | 7836 | yes | 2164
CreditCard | 0 | 2 | 0 | no | 9033 | yes | 967
Gender
FamilyStatus
Profession
SavingsBook
LifeInsurance
CreditCard
OnlineBanking
JointAccount
CashCard
Age
DurationClient
AccountBalance
Save task
Visible fields
14
Charts/row:
0
data records:
10000
Refresh
Export

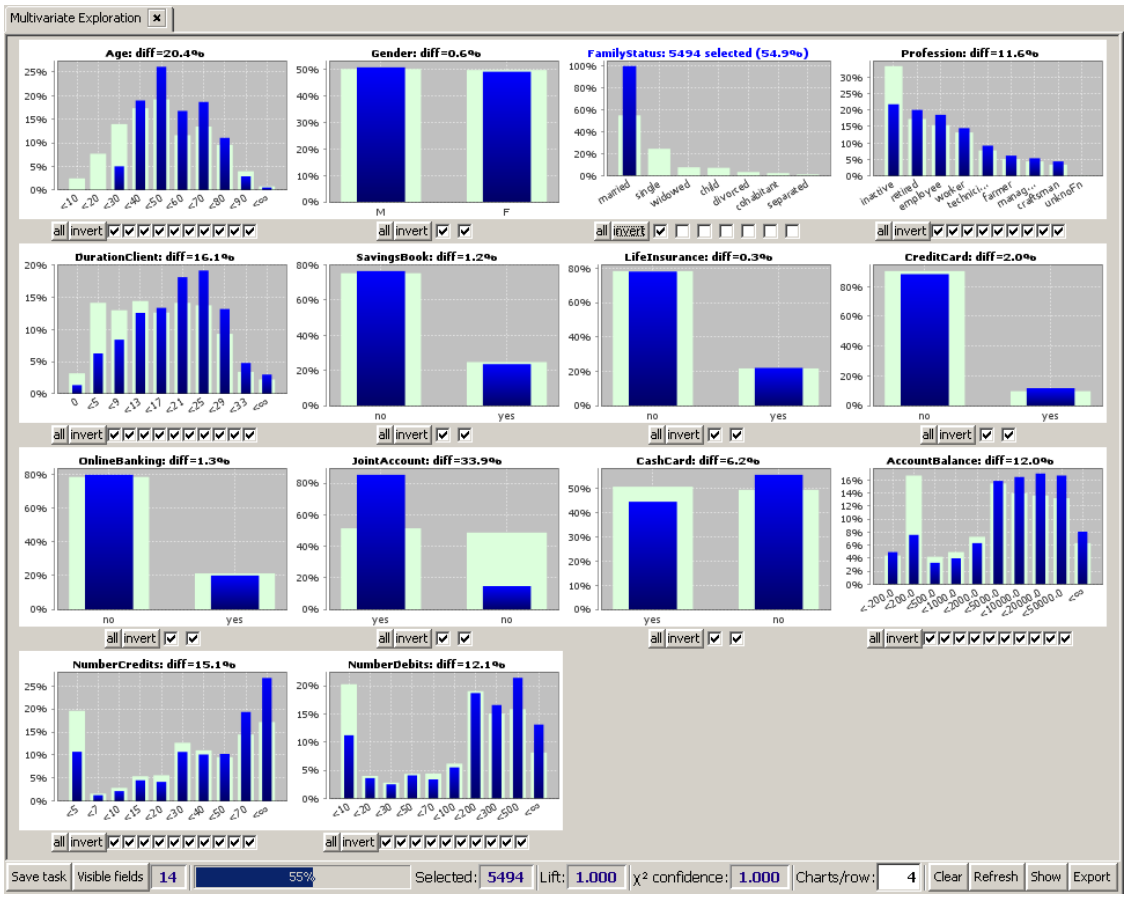
Note: you find more explanations on the module Statistics and Distributions in the module's documentation.

For now, we are satisfied with what we see: the 14 non-group fields apparently contain reasonable values and value distributions, and there are no missing or invalid values. Also, the default binnings and value range definitions performed by Synop Analyzer are suitable for the planned analysis steps. Therefore, we do not further fine-tune the data loading options and directly start a multivariate data exploration.

5.1.6 Step 3: Multivariate Interactive Data Exploration

Pressing the button **Multivariate Exploration** opens a new panel, consisting of one histogram chart per active data field and a tool bar at the lower edge of the tab. Each histogram chart compares a field's value distribution on the currently selected subset (blue bars) to the field's value distribution on the entire data (light green bars).

By clicking on one of the checkboxes below each chart, you can define a value selection for the corresponding data field. For example, in order to select only those customers who are married, click on the leftmost checkmark below the chart for the field `MaritalStatus` (this selects all but the married customers), the click on the **invert** button (this inverts the previous selection, and hence selects only the married customers):



The tool bar at the bottom of the screen shows some overall statistics of the current selection:

- The progress bar on the left and the adjacent text field **Selected** show the size of the currently selected subset of the data: once as a percentage of the entire data, once the absolute number of selected data record.
- The text field **Lift** indicates whether the combination of field value ranges defining the current selection 'attracts' or 'repulses' each other. lift values larger than 1.0 (less than 1.0) indicate that the different selected value ranges occur more (less) frequently together than expected in the case of statistical independence.
- The text field **χ^2-Confidence** contains the statistical confidence that the selected subset differs significantly from the entire data in at least one data field's value distribution. (More formally spoken the value is the confidence level with which the hypothesis 'The currently selected subset has the same value distribution in all data fields as the entire data' is rejected by a χ^2 significance test.)
- The input field **Charts/row** defines how many histogram charts are displayed in one screen row.
- The **Export** button opens a 'save file' dialog which stores a snapshot of the current state of the analysis to a spreadsheet in `.xlsx` file format (MS Excel© 2007+ format).

The fields' histogram charts can have two different appearances, depending on whether or not a value range selection has been performed for the field:

- Fields for which certain values have been selected, others deselected, have a blue title which shows the field name and the number of data records which are covered by the field's current value selection. In the figure above, 5494 records with `MaritalStatus=married` have been selected. Therefore the histogram for the field has a blue title text.
- Fields for which the value range has not been restricted are shown with black title. The title then contains the field name and a percentage number which indicates how much the blue and the light green bars differ, or in other words: how much the field's value distribution on the selected subset differs from the field's value distribution on the entire data. From the figure shown above we learn that married and unmarried customers differ most strongly on the field `JointAccount` (33.9%) and most weakly on the field `LifeInsurance` (0.2%).

The y-scale of each histogram contains a percentage number indicating the relative frequency of the single values or value ranges. For example, looking at the histogram for `Profession` in the picture above we see that on the overall data, about 33% of all data records have `Profession=inactive`, whereas on the currently selected subset (`MaritalStatus=married`) only about 21% of the selected data records have the value `Profession=inactive`.

5.1.7 Step 4: Customer Intelligence with Multivariate Data Exploration

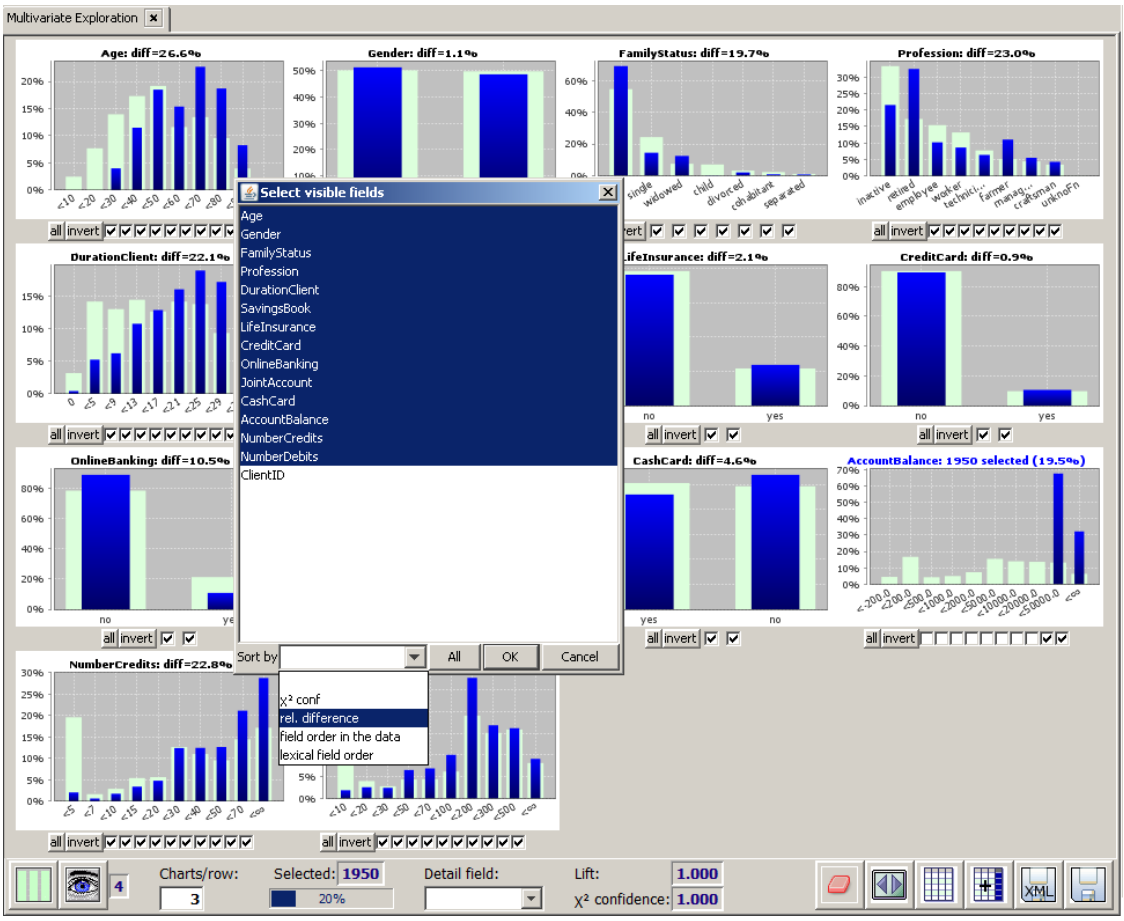
Let us now assume that the multivariate exploration has been started with the goal to analyzing those customers who have large sums of money on their giro bank account. The question is:

- Do these customers have typical attributes? Can they be grouped into some homogenous clusters?
- Which other banking services are they using?
- Are there up-selling potentials for promoting other banking services to those customers?

In order to answer this question, we first undo our current selection by pressing the 'all' button for the field `MaritalStatus` or the **Clear** button in the tool bar.

Then we select only those customers with an `AccountBalance` above 20000: click on the two rightmost checkboxes below the histogram for `AccountBalance`, then click on the **invert** button.

How did this selection of the 1950 customers with the highest average balance score influence the other fields' value distributions? To answer this we re-arrange the histograms by decreasing difference between the selected and the overall data: we click on the **Visible fields** button and then on **Sort by** → **relative difference** (see the picture below).

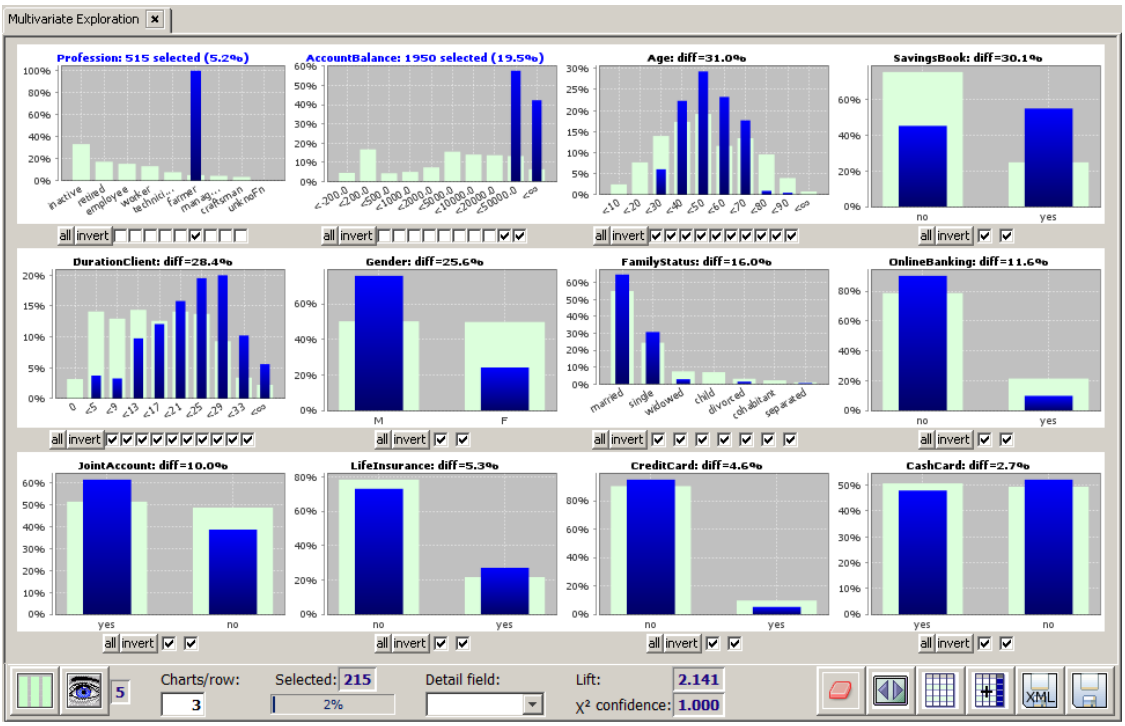


We see that the most significant changes are:

- A shift in the `Age` distribution towards higher ages (not very surprising)
- A significant over-representation of the professions `Profession=pensioner` (also not very surprising) and `Profession=farmer`.

This last observation seems the most interesting to us. We want to focus on those farmers with an account balance of more than 20000 Euros. We therefore narrow down the existing selection by clicking on the fourth checkmark from the right below the histogram for `Profession`, then on the **invert** button. This selects only the value `Profession=farmer`.

Then we again open the **Visible fields** dialog, sort the fields by 'relative difference' and hide the two fields `NumberCredits` and `NumberDebits`, which are of little interest in the analysis we are performing. (Keep the <CTRL> key pressed while clicking on the two field names in order to remove them from the list of visible fields.)



Interestingly, this refinement of the previous selection pushes the age distribution back to the younger age groups (see picture below). Hence, among all customers with high amounts of money on their giro bank account, the farmers rank among the youngest. Furthermore, we see that `Gender=m` and `MaritalStatus=married` and `MaritalStatus=single` are strongly over-represented in this group. And, what is also interesting, most of those 'rich' farmers do not have a life insurance, at least not from our bank. On the other hand, they seem conservative (many of them have a `SavingsBook`, few of them use `OnlineBanking`), and they are very loyal customers (`DurationClient` strongly above average).

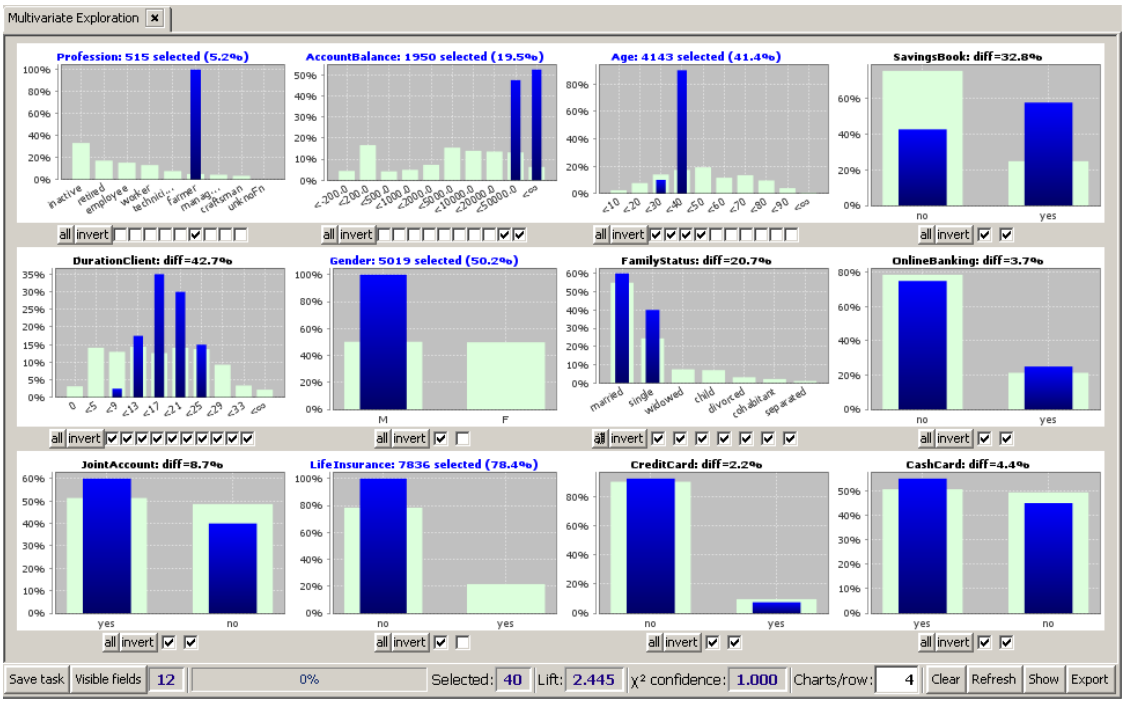
5.1.8 Step 5: Campaign Plannung and Target Group Selection

We believe that in the previous section we have identified a very promising target group for an up-selling campaign. Our idea is that we want to further narrow down our selection to the married male farmers below 40 years who do not yet have a life insurance from our bank. This group seems to be an excellent target group for a phone call or a personal visit with the goal of speaking about a life insurance for protecting the family:

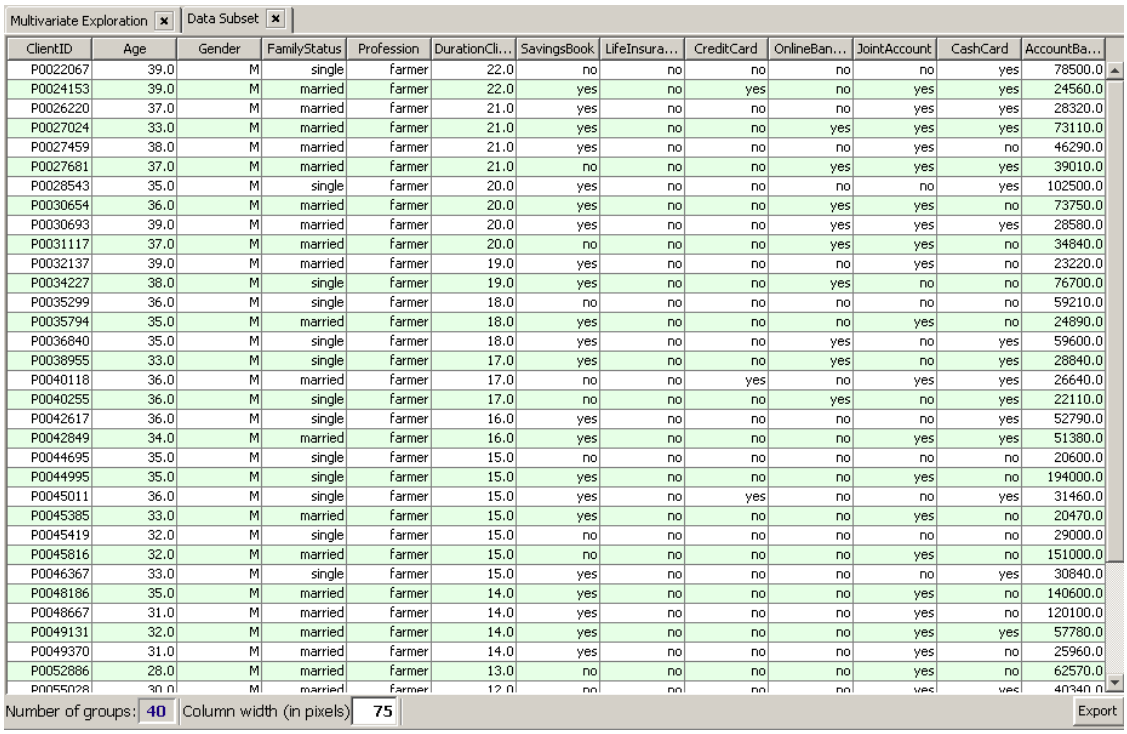
- The group seems to be prosperous.
- The group is relatively young.
- The group is married and probably has children.
- The group has a dangerous profession which demands financial protection of wife and descendants in case of an illness or accident.

- The group has an above-average propensity to have a life insurance.

We perform the selection as described above by clicking on the suitable checkboxes and buttons below the histograms for `MaritalStatus` (married), `Age` (<40), `Gender` (m), `LifeInsurance` (no).

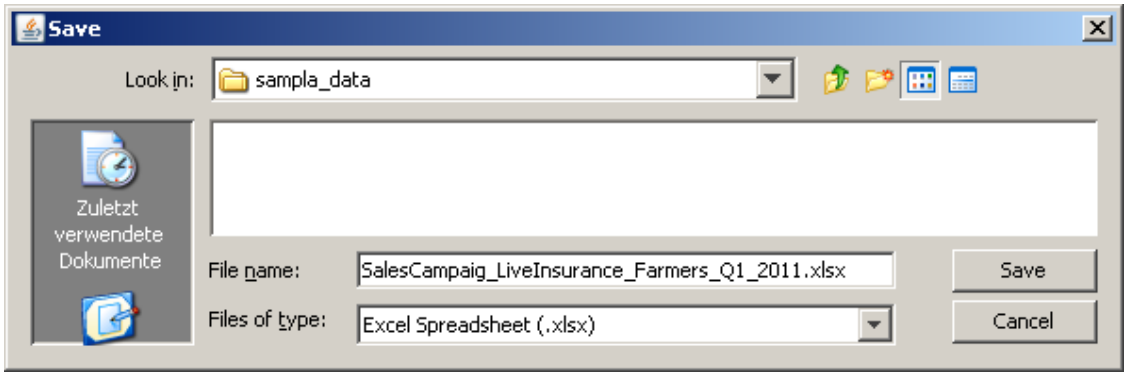


We notice that the remaining group consists of 40 customers, a reasonably small number of customers for being contacted by one sales representative. As a final check before starting the campaign, we would like to introspect the 40 selected data records. To this purpose, we click on the **Show** button on the right end of the tool bar. This brings up a new panel showing the selected data records:

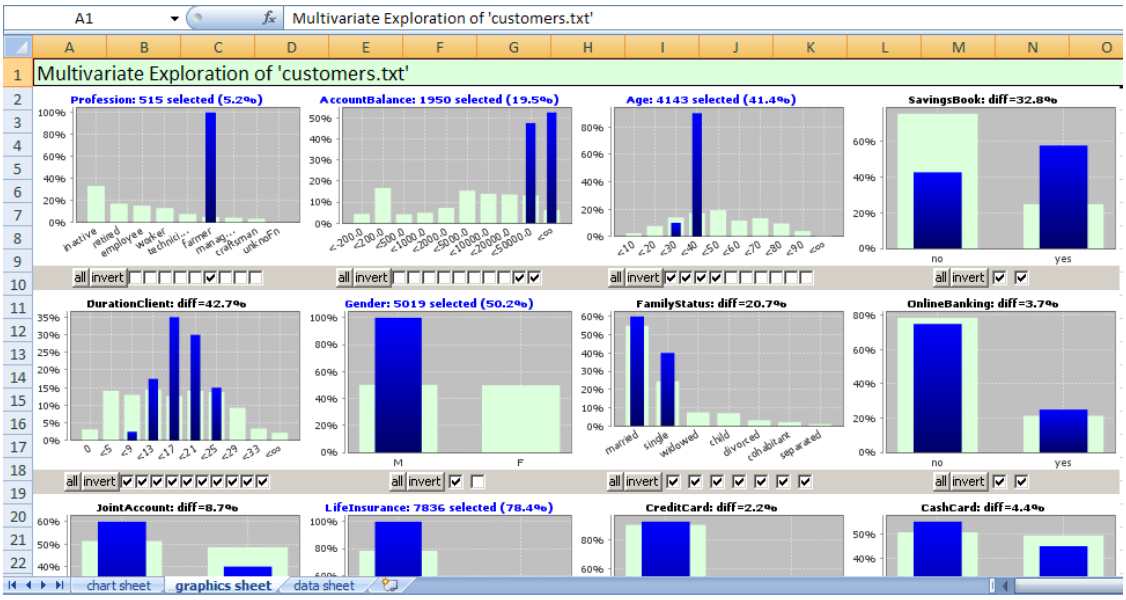


ClientID	Age	Gender	FamilyStatus	Profession	DurationCli...	SavingsBook	LifeInsura...	CreditCard	OnlineBan...	JointAccount	CashCard	AccountBa...
P0022067	39.0	M	single	farmer	22.0	no	no	no	no	no	yes	78500.0
P0024153	39.0	M	married	farmer	22.0	yes	no	yes	no	yes	yes	24560.0
P0026220	37.0	M	married	farmer	21.0	yes	no	no	no	yes	yes	28320.0
P0027024	33.0	M	married	farmer	21.0	yes	no	no	yes	yes	yes	73110.0
P0027459	38.0	M	married	farmer	21.0	yes	no	no	no	yes	no	46290.0
P0027681	37.0	M	married	farmer	21.0	no	no	no	yes	yes	yes	39010.0
P0028543	35.0	M	single	farmer	20.0	yes	no	no	no	no	yes	102500.0
P0030654	36.0	M	married	farmer	20.0	yes	no	no	yes	yes	no	73750.0
P0030693	39.0	M	married	farmer	20.0	yes	no	no	yes	yes	yes	28580.0
P0031117	37.0	M	married	farmer	20.0	no	no	no	yes	yes	no	34840.0
P0032137	39.0	M	married	farmer	19.0	yes	no	no	no	yes	no	23220.0
P0034227	38.0	M	single	farmer	19.0	yes	no	no	yes	no	no	76700.0
P0035299	36.0	M	single	farmer	18.0	no	no	no	no	no	no	59210.0
P0035794	35.0	M	married	farmer	18.0	yes	no	no	no	yes	no	24890.0
P0036840	35.0	M	single	farmer	18.0	yes	no	no	yes	no	yes	59600.0
P0038955	33.0	M	single	farmer	17.0	yes	no	no	yes	no	yes	28840.0
P0040118	36.0	M	married	farmer	17.0	no	no	yes	no	yes	yes	26640.0
P0040255	36.0	M	single	farmer	17.0	no	no	no	yes	no	yes	22110.0
P0042617	36.0	M	single	farmer	16.0	yes	no	no	no	no	yes	52790.0
P0042849	34.0	M	married	farmer	16.0	yes	no	no	no	yes	yes	51380.0
P0044695	35.0	M	single	farmer	15.0	no	no	no	no	no	no	20600.0
P0044995	35.0	M	single	farmer	15.0	yes	no	no	no	yes	no	194000.0
P0045011	36.0	M	single	farmer	15.0	yes	no	yes	no	no	yes	31460.0
P0045385	33.0	M	married	farmer	15.0	yes	no	no	no	yes	no	20470.0
P0045419	32.0	M	single	farmer	15.0	no	no	no	no	no	no	29000.0
P0045816	32.0	M	married	farmer	15.0	no	no	no	no	yes	no	151000.0
P0046367	33.0	M	single	farmer	15.0	yes	no	no	no	no	yes	30840.0
P0048186	35.0	M	married	farmer	14.0	yes	no	no	no	yes	no	140600.0
P0048667	31.0	M	married	farmer	14.0	yes	no	no	no	yes	no	120100.0
P0049131	32.0	M	married	farmer	14.0	yes	no	no	no	yes	yes	57780.0
P0049370	31.0	M	married	farmer	14.0	yes	no	no	no	yes	no	25960.0
P0052886	28.0	M	married	farmer	13.0	no	no	no	no	yes	no	62570.0
P0055028	30.0	M	married	farmer	12.0	no	no	no	no	yes	yes	40340.0

We are satisfied with the selected target group for the sales campaign. Now we want to start the campaign by sending both the analysis rationale and the selected target group to the colleagues who will be responsible for running the campaign but who do not neccessarily have access to Synop Analyzer. We press the **Export** button on the right end of the tool bar of the multivariate analysis panel. A 'Save file' dialog opens up in which we can specify the name of the spreadsheet file into which the analysis results (in the form of png graphics objects) and the selected customer data records will be written.



That file can later be opened in MS Excel© or another front-end by the sales representative who will be in charge of contacting the selected people. It contains two tabs with the analysis summary and one tab with the selected data sets:



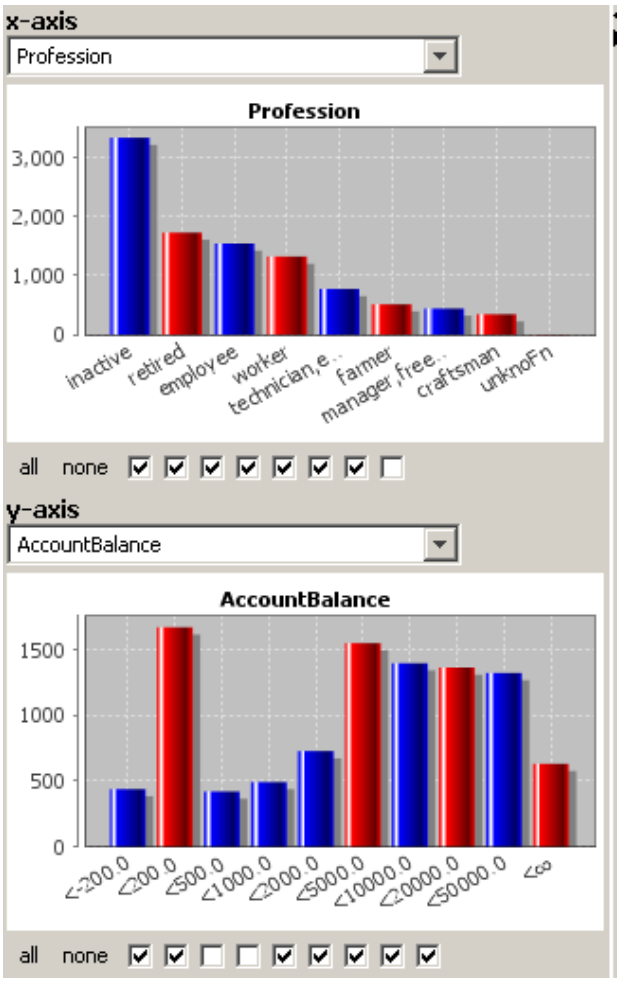
5.1.9 Step 6: Detailed Look at the Interrelation of two Fields

Our successful identification of a subgroup of farmers with a particularly high average balance on their giro accounts motivates us to study the interrelation between a customer's profession and its average balance it more detail.

In the input data panel on the left side of the screen we click on the **Bivariate Exploration** button. A new Bivariate Exploration panel opens up. The new tab is vertically split into two columns. In the left column, you can select the two data fields whose interrelation you want to study. In the right column, you see the resulting bivariate statistics for your selection. Per default, the first two active fields in the data are interrelated. We want to see the fields `Profession` and `AccountBalance` instead. Therefore, we use the two selection fields under 'x-axis' and 'y-axis' and select the fields `Profession` and `AccountBalance`.

By default, each field is divided into two ranges (classes) for the bivariate analysis. For the field PROFESSION that is not what we want to have. Instead, we want to treat each sufficiently frequent profession value as a separate class. Therefore, activate all checkboxes except the rightmost one below the histogram for field `Profession`. This treats the 7 most frequent profession values as separate classes and creates one summary class for the remaining values `craftsmen,...` and `unknown` (see picture below).

The second field we are interested in is the `AccountBalance` field. Here, we activate all but the third and fourth checkbox below the histogram. This creates the value ranges <-200, -200...200, 200...2000, 2000...5000, 5000...10000, 10000...20000, 20000...50000 and ≥50000. (see picture below).



The right-hand side of the panel now shows two bivariate value frequency plots of `AccountBalance` as a function of `Profession`.

The upper chart shows which combinations of `AccountBalance` and `Profession` appear more (green) or less (red) frequently than expected if the two fields were statistically independent.

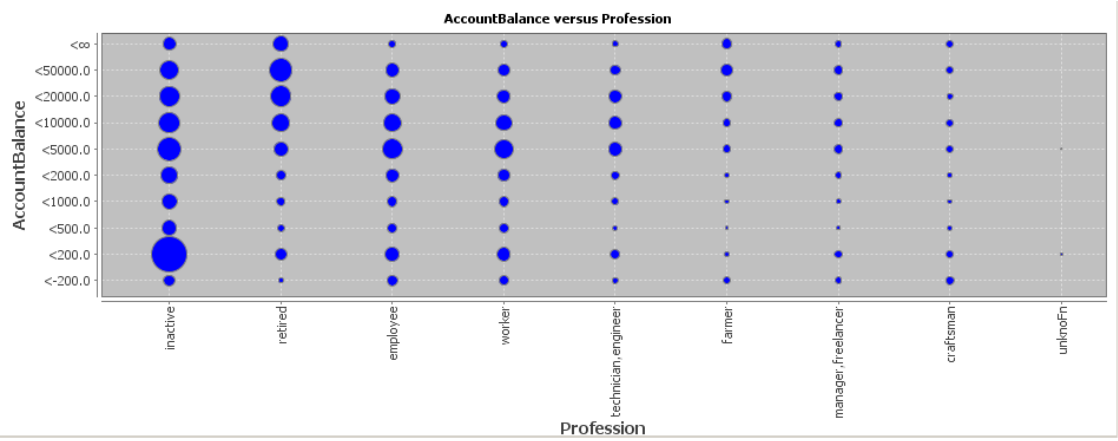
Some of the findings are not very surprising: professionally inactive customers have often an account balance near 0, pensioners and managers often have very high account balances, etc. But there are also some surprising findings, for example that managers are strongly over-represented among the customers who have a significantly negative account balance...</de_DE>Einige der Ergebnisse überraschen nicht wirklich: beruflich inaktive Kunden haben oft einen Kontostand in der Nähe von 0, Rentner und Manager öfter sehr hohe Kontostände etc. Aber es gibt auch Überraschungen, z.B. dass Manager stark überrepräsentiert sind in der Gruppe der Kunden mit signifikant hohem negativen Kontostand...</de_DE>

Profession AccountBalance	inactive	retired	employee	worker	technician,e..	farmer	manager,free..	craftsman, unknoFn	sum	χ² conf
<∞	138 (-34%)	198 (+82%)	45 (-54%)	45 (-46%)	34 (-30%)	91 (+180%)	39 (+39%)	40 (+81%)	630	1.000
<50000.0	281 (-36%)	434 (+90%)	154 (-24%)	123 (-29%)	91 (-11%)	124 (+82%)	69 (+18%)	44 (-5%)	1320	1.000
<20000.0	344 (-24%)	356 (+51%)	202 (-4%)	145 (-19%)	132 (+25%)	90 (+28%)	63 (+4%)	30 (-37%)	1362	1.000
<10000.0	357 (-23%)	263 (+9%)	262 (+22%)	214 (+16%)	144 (+33%)	55 (-23%)	59 (-5%)	41 (-16%)	1395	1.000
<5000.0	438 (-15%)	169 (-37%)	325 (+36%)	292 (+43%)	160 (+34%)	55 (-31%)	66 (-4%)	42 (-23%)	1547	1.000
<2000.0	622 (+14%)	173 (-39%)	293 (+16%)	279 (+29%)	114 (-10%)	40 (-53%)	67 (-8%)	51 (-11%)	1639	1.000
<200.0	1047 (+89%)	112 (-61%)	174 (-32%)	148 (-33%)	71 (-45%)	23 (-73%)	44 (-41%)	49 (-16%)	1668	1.000
<-200.0	101 (-31%)	22 (-71%)	86 (+27%)	74 (+28%)	28 (-18%)	37 (+64%)	37 (+90%)	54 (+250%)	439	1.000
sum	3328	1727	1541	1320	774	515	444	351	10000	
χ² conf	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000		1.000

The black 'sum' row and column respectively contains the total number of data records which have the field value given in the column or row header. The black number in the right corner is the total number of records (or, if the check box 'ignore invalid/missing values' in the left part of the panel has been selected, the total number of records which have a valid value in both considered fields.

The meaning of the **χ^2-conf** columns and rows as well as some other more advanced features of the bivariate exploration modules are explained in a separate section of the documentation.

The lower part of the right column contains a bivariate plot of absolute value pair counts, the area of each blue circle being proportional to the represented number of records which have the field values at the position (x,y):



This kind of plot helps in identifying quickly the 'hot spots' of the value pair distribution, i.e. the most frequent field value combinations.

5.1.10 Summary

In Summary, we have demonstrated how customer data can be intuitively explored using Synop Analyzer. The exploration required neither an elaborate data preprocessing nor

sophisticated statistical or tool handling skills and created insights and evidence which can be immediately applied for marketing campaign planning, sales controlling and other management tasks.

6

Glossary

χ^2 conf
(module: **Bivariate Exploration and Correlations***)*
χ^2 confidence indicates whether or not the field value distribution of one field significantly changes when the other field has a specific value or a value in a specific range. χ^2 confidence numbers are numbers between 0 and 1. The closer to 1, the higher the statistical evidence that a significant impact of one field on the value distribution of the other field has been detected. In general, statisticians consider an impact as 'significant' if the χ^2 confidence exceeds a value of 0.95 ('95% confidence level') or 0.99 ('99% confidence level') A χ^2 confidence number appearing as the rightmost number of a normal matrix row indicates whether the value distribution of the x-axis field systematically differs from its general behavior if the y-axis field assumes the value or value range which is indicated in the leftmost entry of that row. A χ^2 confidence number appearing as the last number of a normal matrix column indicates whether the value distribution of the y-axis field systematically differs from its general behavior if the x-axis field assumes the value or value range which is indicated in the first entry of that column. The χ^2 confidence number in the bottom-right matrix corner indicates whether there is a significant dependence of the x-axis field's value distribution from the y-axis field's value and vice versa.

χ^2 conf
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that the value distribution of the selected data subset differs in a statistically significant way from the overall data's value distribution on the currently selected data field. The confidence is calculated based on the confindence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

χ^2 conf
(module: **Multivariate Exploration and Split Analysis***)*

The confidence that the value distributions of the test and the control data differ in a statistically significant way in at least one of the data fields in which the control data are not selected manually but chosen automatically to be as similar to the test data distribution as possible. The confidence is calculated based on the confidence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

χ^2 conf
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that the value distributions of the test data and the contol data differ in a statistically significant way on the currently selected data field. The confidence is calculated based on the confindence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

χ^2 conf.
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that the overall value distribution of the selected subset differs in a statistically significant way from the overall value distribution on the entire data. The confidence is calculated based on the confidence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

χ^2 conf.
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that deviation of the overall selection's lift from 1 is statistically significant. The confidence is calculated based on a χ^2 significance test with one degree of freedom.

χ^2 confidence
(module: **Associations Analysis***)*
The χ^2 confidence level of an association indicates up to which extent each single item is relevant for the association because its occurrence probability together with the other items of the association significantly differs from its overall occurrence probability. More formally, theχ^2 confidence level is the result of performing n χ^2 tests, one for each item of the association. The null hypothesis for each test is: the occurrence frequency of the item is independent of the occurrence of the item set formed by the other n-1 items. Each of the n tests returns a confidence level (probability) with which the null hypothesis is rejected, and the χ^2 confidence level of the association is set to the minimum of these n rejection confidences.

Abs. support
(module: **Associations Analysis***)*
The absolute support of an association is the number of groups (transactions) in which the association occurs. When specifying the parameters for an associations training, you

should always specify an lower boundary for the absolute or relative support, otherwise the training can take extremely long time.

Abs.diff.
(module: **SOM Models***)*
Maximum absolute difference to the field's overall value distribution: the SOM card shows the nominal value for which the difference between its actual frequency within the records mapped to the given neuron and its expected frequency is maximum.

Absolute support
(module: **Sequential Patterns***)*
The absolute support of a sequence is the number of entities in which the sequence occurs.

Additive Season
(module: **Time Series Analysis***)*
Additive season means that the seasonal pattern is modeled as an added term to the long-term trend ('total = trend + season'). As a result, the amplitude of the seasonal fluctuations is constant and does not grow when the trend line increases. Multiplicative season means that the seasonal pattern is modeled as a correction factor to the long-term trend ('total = trend * season'). As a result, the amplitude of the seasonal fluctuation increases when the trend line increases and decreases when the trend line decreases.

Allow irreversible binning
(module: **Data Import***)*
If this check box is marked, numeric data fields can be discretized into a small number of intervals, and the original field values are irreversibly replaced by interval indices. For example, the value AGE=37 might be replaced by AGE=[30..40[, and in the compressed data, the precise value 37 will be irreversibly lost.

Assoc Model
(modules: **Workbench, Data Import, Associations Analysis***)*
An associations model is a collection of association rules which have been detected during an associations training run on the training data set. In the associations model panel, you can visualize and introspect the results of an associations training run. You can display the results in tabular form, sort, filter and export the filtered results to flat files or into a table in a RDBMS. Furthermore, you can calculate additional statistics for the support of single associations in the introspected result.

Associations Detection
(modules: **Workbench, Data Import, Associations Analysis***)*
In this module, you specify the parameters and settings which are to be used for the next associations training run. Furthermore, you can store your parameter settings, manage them in a repository and later retrieve and reuse them. In the lower part of the panel, you

can start and stop an associations training run and monitor its progress and its predicted run time.

Associations Scoring
(modules: **Workbench, Data Import, Associations Analysis***)*
An associations scoring matches a collection of association rules (an associations model) with a new data table and indicates which associations are fulfilled (supported) by which data sets. In the associations scoring task panel, you specify the parameters and settings which are to be used for applying detected associations to new data or for gathering additional statistics on the supporting transactions of certain associations. You can store your parameter settings, manage them in a repository and later retrieve and reuse them. In the lower part of the panel, you can start and stop associations application runs and monitor their progress and predicted run time.

Automized data field
(module: **Multivariate Exploration and Split Analysis***)*
Data field over whose values an automatically executed series of split analyses is to be performed. Automizable data fields are all fields on which one single value has been selected on the test data and several other values have been selected on the control data. During each step of the automized series analysis, a different single value out of the initially selected test and control data values is considered the test data and all remaining initially selected values the control data.

Boolean field
(module: **Data Import***)*
A data field which is to be treated as Boolean field. If it contains more than 2 different values, all but the the first two different values will be ignored, i.e. treated as missing values.

Browser call
(module: **Workbench***)*
For accessing online help, the software must start an external web browser. This parameter contains the calling command for this browser. There are default settings for several operating systems. Therefore, you should only modify this parameter if you are unable to use the online help with the default settings.

Buffer page size
(module: **Workbench***)*
The data page size (in bytes) which is used in the preliminary representation of data field objects. Allowed values are 10000 to 10000000. Larger values can speed up the data reading, but they can also raise memory requirements, in particular on data with many fields.

Cancel the training
(modules: **Associations Analysis, Sequential Patterns, SOM Models, Regressions Analysis, Decision Trees***)*
Aborts the currently running training task without creating a result.

Chart start
(module: **Time Series Analysis***)*
First time point shown in the time series charts

Chart width (pixels)
(modules: **Statistics and Distributions, Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
The resolution (number of pixels in x direction) of the single histogram charts. The number refers to 'normal' charts. Extra-wide charts withmany histogram bars have a resultion which is a multiple of this number.

Charts/row
(modules: **Statistics and Distributions, Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
The Number of histogram charts per row. If this value is 0, the software automatically selects a suitable number of charts per row, depending on the total number of charts to be shown.

Child support ratio
(modules: **Associations Analysis, Sequential Patterns***)*
Specify a lower boundary for the acceptable 'support shrinking rate' when creating expanded associations out of existing associations. An expanded association of n items will be rejected if at least one of the n parent associations has a support which is so large that when multiplied with the minimum shrinking rate, the result is larger than the actual support of the expanded association.

Chi^2 conf
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that the value distributions of the test and the control data differ in a statistically significant way in at least one of the data fields in which the control data are not selected manually but chosen automatically to be as similar to the test data distribution as possible. The confidence is calculated based on the confidence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

Chi^2 conf
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that the value distributions of the test data and the contol data differ in a statistically significant way on the currently selected data field. The confidence is

calculated based on the confindence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

Chi^2 confidence (in the toolbar at the bottom edge of the panel)
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that the overall value distribution of the selected subset differs in a statistically significant way from the overall value distribution on the entire data. The confidence is calculated based on the confidence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

Chi^2 confidence (in the toolbar at the bottom edge of the panel)
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that deviation of the overall selection's lift from 1 is statistically significant. The confidence is calculated based on a χ^2 significance test with one degree of freedom.

Chi^2 confidence (of an association pattern)
(module: **Associations Analysis***)*
The χ^2 confidence level of an association indicates up to which extent each single item is relevant for the association because its occurrence probability together with the other items of the association significantly differs from its overall occurrence probability. More formally, theχ^2 confidence level is the result of performing n χ^2 tests, one for each item of the association. The null hypothesis for each test is: the occurrence frequency of the item is independent of the occurrence of the item set formed by the other n-1 items. Each of the n tests returns a confidence level (probability) with which the null hypothesis is rejected, and the χ^2 confidence level of the association is set to the minimum of these n rejection confidences.

Chi^2 confidence (within a histogram chart title)
(module: **Multivariate Exploration and Split Analysis***)*
The confidence that the value distribution of the selected data subset differs in a statistically significant way from the overall data's value distribution on the currently selected data field. The confidence is calculated based on the confindence level with which the null hypothesis 'the two value distributions are identical' is rejected by a χ^2 test.

Computed fields
(module: **Data Import***)*
Define additional data fields whose values are to be computed from the values of one or more existing data fields.

Confidence
(modules: **Associations Analysis, Sequential Patterns***)*
The confidence of an association rule or sequence rule is the ratio between the rule's support and the rule body's support. An association rule is an association of n items

in which n-1 of the n items are considered the 'rule body' and the remaining item is considered the 'rule head'. Hence, n different association rules can be constructed from one association of length n. Similarly, a sequence rule is a sequence of n sets of items - separated by n-1 time steps - in which the first n-1 item sets are considered the rule body and the item set after the last time step is considered the rule head. A rule's confidence is the probability that the rule head is true if one knows for sure that the rule body is true.

Confidence range
(module: **Pivot Tables***)*
This value determines whether an error bar (confidence range) is to be drawn for each point in the diagram, and it determines the confidence range represented by the error bar.If the confidence value C is selected here, that means that a positive or negative deviation from the actual value in y-direction is with a confidence of C due to a significant change in the probability distribution and can not be explained by just a statistic fluctuation within the current probability distribution.

Confidences
(modules: **Associations Analysis, Sequential Patterns***)*
The confidences C of the n different ways of interpreting the association as a rule of the form 'if (itemX and itemY and ... are present in a transaction) then also itemZ is present in the transaction with a probability (confidence) of C, in short notation: (itemX,itemY,...) =(C)=> itemZ. The first number in the list corresponds to the rule (item2,item3,...) =(C)=> item1, the second to the rule (item1,item3,...) =(C)=> item2, and so on.

Confidences
(module: **Sequential Patterns***)*
The confidences C of the n consecutive steps of the sequence. The first number in the list is the probability that an arbitrary entity contains the first item set of the sequence. The second number is the probability that an entity containing the first set also contains the sequence's second item set, and so on.

Contingency
(module: **Bivariate Exploration and Correlations***)*
Cramer's contingency coefficient V as described in http://en.wikipedia.org/wiki/Contingency_table

Control data
(module: **Multivariate Exploration and Split Analysis***)*
The currently selected control data subset in a test-control data analysis. The goal of the analysis is to detect and quantify systematic deviations in the field value distribution properties between the test data subset and the control data subset

Core item purity
(module: **Associations Analysis***)*
The core item purity of an association is the ratio between the association's support and the support of the least frequent item within the association. A core item purity of 1 indicates a 'mononuclear' group in which the support of the group is determined by the support of its least frequent item. Note: the core item purity is always larger than or equal to the association's purity.

Correction Hints
(modules: **Deviation Detection, Associations Analysis***)*
A set of possible corrections which would help removing an inconsistency from some data records. The hints are created based in a statistical analysis of the involved items.

Correlations Analysis
(modules: **Workbench, Data Import, Bivariate Exploration and Correlations***)*
The correlation between two data fields indicates whether or not there is a significant statistical dependency between the values of the two data fields. The correlations module computes and visualizes these field-field correlations

Create a new residual field in the data
(module: **Regressions Analysis***)*
Create a new field in the input data which contains the residuals 'actual target value - predicted target value'. The name of the new field is [targetFieldName]_RESIDUAL.

Create persistent data file
(module: **Data Import***)*
If this check box is marked, a persistent version of the compressed data object will be written to a file and can be refetched later. This speeds up the data reading process in future mining sessions on this data object.

Data block size
(module: **Workbench***)*
Data block size (in bytes) in block-wise data reading from flat text files. allowed values are 100000 to 1000000000

Data groups
(module: **Statistics and Distributions***)*
Number of different data groups (group field values) in the input data

Data Subset
(module: **Alle***)*
In this panel, you can explore the data selections created by a multivariate data explotation or another data analysis module.

Data to be joined-in
(module: **Data Import***)*
Name of the data source from which certain fields are to be added to the currently active main data source

Default result directory
(module: **Workbench***)*
Default directory path in which analysis results are stored.

Detail field
(module: **Multivariate Exploration and Split Analysis***)*
Name of the data field whose value distribution defines the colors of the histogram bars representing the selected data set. When no detail field is selected, the histogram bars are displayed without detail structure and in uniformly blue color.

Detail field
(module: **Time Series Analysis***)*
For each value of this field, a separate time series chart will be drawn

Deviation strength
(module: **Deviation Detection***)*
The strength of a deviation pattern describes how strongly and significantly the number of occurrences of the pattern is below the expected number of occurrences.The value is calculated as '10*(chi^2-conf - 0.9) / lift', where 'lift' is the pattern's lift and 'chi^2-conf' is the confidence level that the pattern is statistically significant.For example, if a combination (A,B) of two data field values A and B occurs in 0.02% of all records and has a chi^2 confidence level of 0.99, and if A and B alone occur in 20% respectively 10% of the data records, then the deviation strength of the pattern (A,B) is 90 since lift is 0.02%/(20%*10%) = 1/100 and 10*(chi^2-conf - 0.9) = 0.9.

Deviations, Inconsistencies
(modules: **Workbench, Data Import, Deviation Detection***)*
In the Deviation Detection panel, outliers, deviations and presumable data inconsistencies can be detected..

Diff. values
(module: **Statistics and Distributions***)*
Number of different valid values of the data field. Note: for binned numeric fields, only those different values are counted which were encountered while collecting statistics for determining the bin boundaries.

Difference
(modules: **Multivariate Exploration and Split Analysis, Multivariate Explo-**

ration and Split Analysis*)*
Difference: #selected - #expected.

Discrete field
(module: **Data Import***)*
A data field which is to be treated as discrete numeric field. If it contains textual values, these values will be ignored, i.e. considered as missing values.

Empty field threshold
(module: **Workbench***)*
Data fields in which (almost) no data row has a valid value are normally of little interest within a data analysis. Therefore the software drops these fields when reading data from a data source. The parameter 'Empty field threshold' specifies the minimum filling rate below which a field will be dropped. The minimum filling rate is a number between 0.0 and 1.0; it describes the fraction of all data records in which the field has a valid value.

Entities
(module: **Statistics and Distributions***)*
Number of different entities (entity field values) in the input data. If no entity field has been specified, the number of entities is equal to the number of groups, or, if no group field has been specified, equal to the total number of data records.

Entity field
(module: **Data Import***)*
Specify a data field which marks several adjacent data records as referring to one single entity (such as a customer, a car, a product, or a patient). The entity data field contains the entity identifier (such as a customer or vehicle or product or patient ID).

ES alpha
(module: **Time Series Analysis***)*
Exponential Smoothing coefficient alpha. (defines a damping factor (1-alpha) per time step.

ES weight
(module: **Time Series Analysis***)*
Weight prefactor to the Exponential Smoothing part of the forecast; weight=0 switches off the Exponential Smoothing.

Excess
(module: **Statistics and Distributions***)*
The sample excess of the value distribution. Note: the sample excess slightly differs from population excess (e.g. MS Excel's 'Excess Curtosis').

Expected
(module: **Multivariate Exploration and Split Analysis***)*
Expected number of data records or data groups in the selected data subset, assuming that the field value distribution on the selected data is identical to the field value distribution on the entire data.

Expected number of selected data records
(module: **Multivariate Exploration and Split Analysis***)*
Expected number of data records or data groups in the selected data subset, assuming that the field value distribution on the selected data is identical to the field value distribution on the entire data.

Explained fraction of target variance (R^2)
(module: **Regressions Analysis***)*
R^2 is a measure for the predictive power of the regression model. R^2 near 1 means that the model is able to predict the target values almost perfectly, R^2 near 0 means that the model is almost useless.

Export the compressed data object
(module: **Workbench***)*
Save the in-memory data object as persistent iad file.

Export the data into a text file
(module: **Workbench***)*
Export the data to a data table or flat file, preserving the all settings such as active field definitions, field types, discretizations, name mappings or joined tables. For data with set-valued fields or with a group field, you can choose among several output data formats: The 'set-valued' format: one data row per group; all values of set-valued attributes are written into one single textual string within curly braces \\ and separated by comma. The 'pivoted' format: several data rows per group; all attributes are put into one single 'item' column, which contains values of the form [ATTRIBUTE_NAME]=[VALUE]. The 'boolean fields' format: one data row per group; for each textual value of each non-numeric attribute, the exported data contains one separate Boolean attribute containing '1' if the corresponding attribute value occurs in the current group, and '0' if it does not. The 'only group IDs' format: creates a one-column output in which only the group IDs of the current data set are contained. This format is helpful if the exported data is only aimed to serve as a list of unique keys describing a subset of data records form a larger table.

Field containing the mapped values
(module: **Data Import***)*
The data field in the auxiliary table which contains mapped names for the original values of the affected data field in the main table

Field containing the original values
(module: **Data Import***)*
The data field in the auxiliary file or table which contains the different original values which also appear in the main table field for which the name mapping is being defined. Often, this field is a primary key field of the auxiliary table.

Field containing the taxonomy parents
(module: **Data Import***)*
The data field in the auxiliary file or table which contains the group or category values

Field discretizations
(module: **Data Import***)*
A discretization defines a 'binning' or 'grouping' of fine grained information from a numeric or textual data field into a small number or classes. For textual fields, this means that only the N most frequently appearing textual values will be treated as separate values. All other values are represented by the group 'others'. For numeric fields, this defines a binning into N value ranges (intervals). The interval boundaries are chosen automatically. If the automatically determined interval boundaries for a numeric field are not satisfying, user-defined interval boundaries can be specified manually by entering a list of N-1 numbers, time or date values in ascending order.

Fields to be added
(module: **Data Import***)*
Data fields from the added data source which are to be joined into the currently active main data.

File or table containing the name mappings
(module: **Data Import***)*
A flat file or database table containing at least two data fields (columns). One column contains the different values which currently appear in the main table's data field for which the name mapping is to be defined. The second column contains a mapped value for each of the original different values.

File or table containing the taxonomy relations
(module: **Data Import***)*
A flat file (i.e. column separated text file) or database table which contains at least two data fields (columns): a 'parent' column and a 'child' column. The 'parent' and 'child' values in each data row describe one single hierarchy relation between a group or category (parent) and a member of the group or category (child).

Forecast start
(module: **Time Series Analysis***)*

Starting time point for calculating the aggregated forecast values which are shown below the title line of each chart in the time series forecast screen.

Forecasts
(module: **Time Series Analysis***)*
Number of future time series data points to be forecasted

Foreign key field
(module: **Data Import***)*
A data field which is the primary key of another data file or table. The data field can be used to join that other data table into the current data source.

Freq.
(module: **SOM Models***)*
Maximum frequency: the SOM card shows the nominal value which is the most frequent value on the data records mapped to the given neuron.

Frequency
(module: **Data Import***)*
This parameter defines a lower boundary for the number of data records or data groups on which a value of a non-numeric data field must occur for being tracked as a separate field value and a separate bar in histogram charts. Less frequent values will be grouped into the category 'others'.

Frequency threshold for perfect tupels
(module: **Workbench***)*
Default setting for the minimum required frequency above which a tupel of several items can be considered as a perfect tupel. Must be an integer larger than 1.

Graphs per row
(module: **Time Series Analysis***)*
Number of time series graphs per row

Group field
(module: **Data Import***)*
The input data for data mining can be pivoted or unpivoted. In the unpivoted data format, each 'object to be analyzed' (for example a customer, a process or a production tranche) is represented by exactly one data record (data row). In this case, no group column has to be specified. In the pivoted data format, each 'object to be analyzed' can span multiple adjacent rows of data: there is one 'item' column containing one single property of the object per data row; and there is a 'group' column which contains an unambiguous identifier for the object to which the current data row belongs. In this case, the name of that group column must be specified here. One such 'object to be analyzed' is often called a 'transaction'.

Height of the neural net
(modules: **SOM Models, Reporting***)*
The number of neurons in direction y. Should be a number between 2 and 100

Height-width ratio
(module: **Time Series Analysis***)*
Height to width ratio of the time series charts to be created.

Icon (large)
(module: **Workbench***)*
The icon to appear in the 'Help'–>'About' info screen. When working without a license key (free test version), you can freely change that icon. When working with a license key, the license key checks that the name of the icon corresponds with the information stored in the lichense key.

Icon (small)
(module: **Workbench***)*
The icon to appear in the upper left corner of the graphical workbench window. When working without a license key (free test version), you can freely change that icon. When working with a license key, the license key checks that the name of the icon corresponds with the information stored in the lichense key.

Ignore invalid/missing values
(module: **Bivariate Exploration and Correlations***)*
Ignore all missing and invalid values in the bivariate analysis.

Include constant offset term
(module: **Regressions Analysis***)*
If this check box is marked, a linear model with constant term (y = b0 + b1*x1 + ... + bn*xn) will be created. Otherwise, a model without the term b0 will be created.

Incompatible items
(modules: **Deviation Detection, Associations Analysis, Sequential Patterns***)*
If a set of items has been specified as incompatible (by pairs), then none of the detected deviations, associations or sequences will contain more than one item out of this set. Enter several patterns, separated by comma (,). If a pattern contains a comma as part of the pattern name, escape it by a backslash (\). Each pattern can contain one or more wildcards (*) at the beginning, in the middle and/or at the end.

Index
(module: **Statistics and Distributions***)*
Value index, i.e. the value's position on the list of all values. For numeric fields, value indices are assigned in the natural order of the values: the smallest value has inde 1. For textual fields, value indices are assagined by decreasing frequency: the most frequent

value of a data field has the index 1, the second most frequent one the index 2 and so on.

Initial learning rate
(module: **SOM Models***)*
A number between 0 and 1 which indicates how much the input weights of the best matching neuron are moved towards the field values of a data record when that record is presented to the SOM net during training.

Input Data
(module: **Workbench***)*
In this panel, you can define, describe, preprocess and manage a data source that you want to use for the subsequent data analysis steps.

Intersection
(module: **Associations Analysis***)*
If 'superset' is checked, the 'Show', 'Explore' and 'Export' buttons will handle each data record or group which supports at least one of the selected associations. If 'intersection' is checked, the 'Show', 'Explore' and 'Export' buttons will only handle those data groups which support *all* selected associations.

Interval bounds (numeric fields only)
(module: **Data Import***)*
Specify the desired interval boundaries. Specify n-1 numeric values in ascending order, separated by ';', '|' or ' ' for obtaining n intervals.

Invalid or NULL
(module: **Statistics and Distributions***)*
Number of data records (resp. data groups in the pivoted data format) in which the data field has no valid value.

Invert
(modules: **Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
Invert the field value selection on the current data field in a Multivariate exloration or a test-control data analysis: deactivate the previously selected value ranges and activate those ranges which were filtered out.

Item
(modules: **Deviation Detection, Associations Analysis, Sequential Patterns***)*
An item is an atomic part of an association or sequential pattern, i.e. a single piece of information, typically of the form [field name]=[field value] or [field name]=[field value range from ... to ...].

Item frequencies
(modules: **Associations Analysis, Reporting***)*
The absolute supports of the single items within the association (the first number corresponds to item1, the second to item2, etc.) A star (*) after the number indicates that the item belongs to the core of the association. The core of an association is the smallest possible subset of items of the association which has the same support as the entire association.

Item pair purity
(modules: **Associations Analysis, Sequential Patterns***)*
The item pair purity of two items i1 and i2, is the number of transactions in which both items occur divided by the maximum of the absolute supports of the two items. Item pairs with a purity of 1 are 'perfect pairs': whenever i1 occurs in a transaction, also i2 occurs in it, and vice versa.

Item set length
(module: **Sequential Patterns***)*
The desired item set lengths in the sequences to be detected. Each 'equal-time' part of a sequence is an item set. In the sequence [A] »> [B],[C],[D], for example, the minimum item set length is 1, the maximum item set length is 3.

Item supports
(modules: **Deviation Detection, Associations Analysis, Sequential Patterns***)*
Number of data records or groups on which the different items which form the pattern appear.

JDBC connection string
(module: **Data Import***)*
The string which is sent to a database management system (DBMS) for getting access to a data table via the JDBC protocol. The string contains the DBMS name, hostname and database name. A default version of this string is automatically created from the user's input for DBMS type, host name and database name in the database connect panel. If this default string does not work properly, the manual specification of ':[4-digit port number]' after the host name might be necessary.

Joined tables
(module: **Data Import***)*
Define tables and fields within them which are to be joined into the main table - for example master data tables containing additional properties of certain field values of the main table.

Key field in joined file
(module: **Data Import***)*

Key field in the added data source, must contain the same values as the foreign key column in the main data.

Key-like field threshold
(module: **Workbench***)*
Textual fields which contain a very large number of different values are interpreted as 'key-like' fields; the software assumes that their content is not suitable for being incorporated into subsequent analysis or data mining steps, and they are dropped when reading the data source. This parameter defines the number of different field values above which a field is classified as 'key-like'. Allowed values are 100 to 1000000.

Language
(module: **Workbench***)*
Language in which all textual elements of the graphical workbench will appear

Last point completion
(module: **Time Series Analysis***)*
Completion rate of the last time point, compared to the earlier time points. If, for example, each time point describes the sales figures of one month and for the current month, the current number only covers the accumulated sales figures of 5 out of 25 sales days, then the completion rate of the last time point should be set to 0.2.

License key file
(module: **Workbench***)*
File containing the license key for the software. The file name starts with IA_license_-key. There is no license key file if you are working with a free test or trial version of the software.

Lift
(modules: **Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
This measure compares the actual number of data groups passing the selection criteria to the expected number which would arise if all data fields used as selection criteria were statistically independent. A lift value larger than 1 indicates that the field values used as selection criteria 'attract' each other, a value smaller than 1 indicates that the field values 'repulse' each other.

Lift
(module: **Associations Analysis***)*
The lift of an association is the actual relative support of the association divided by the product of the relative supports of the items which form the association. Associations with lift>1 are 'frequent patterns': the items within the association occur more frequently together than expected if these items were statistically independent. Associations with

lift<1 are 'exceptions' or 'deviations': the items within the association occur less frequently together than expected if these items were statistically independent.

Lift
(module: **Sequential Patterns***)*
The lift of a sequence is a measure for the positive correlation of the item sets (events) which form the sequence. Sequences with lift>0.5 are 'frequent patterns': the item sets within the sequence occur more frequently in that order than expected if the items were statistically independent. Sequences with lift values close to zero are 'exceptions' or 'deviations': the items within the sequence occur less frequently in that order than expected if the items were statistically independent.

Lift increase factor
(module: **Associations Analysis***)*
An association of n items has n lift increase factors, namely the n ratios of this association's lift divided by the lifts of its n different 'parent' associations. A parent association is an association which results when one of the n items is dropped. Specifying limits for the lift increase factor helps keeping the result size manageable by suppressing the generation of redundant child patterns for significant parent patterns. When searching for frequent patterns, lift increase factors greater than 1 should be applied, e.g. 1.5. When searching for deviations, lift increase patterns smaller than 1 should be applied, e.g. 0.5. As an example, let us consider the association ('AGE<18' and 'FAMILY_STATUS=child'). On real-life demographic data, this association is a typical frequent pattern with a lift largely above 1, e.g. 3.62. Therefore, when searching for frequent patterns with lift>3, this pattern will be detected. However, most likely also the following patterns will be detected: ('AGE<18' and 'FAMILY_STATUS=child' and 'GENDER=male'), ('AGE<18' and 'FAMILY_STATUS=child' and 'GENDER=female' and 'STATE=CA'), and many more. All these extended patterns most probably have a lift very close to 3.62 since the pattern extensions are just adding uncorrelated information to the significant 'core' pattern ('AGE<18' and 'FAMILY_STATUS=child'). Setting a minimum lift increase factor of 1.5 helps suppressing all these useless extensions as none of them has a lift greater than 5.43 = 1.5*3.62.

Lift increase factor
(module: **Sequential Patterns***)*
The lift increase factor relates the lift of a sequence to the lift of its parent sequences which results from removing one single item from one of the n equal-time item sets of the sequence. Specifying limits on the lift increase factor helps suppressing the generation of redundant, uninteresting sequences for interesting 'core' sequences. For more detail refer to the explanation of lift increase factor in the associations training module.

Linear
(module: **Regressions Analysis***)*
In linear regression, the value of a numeric target field t is expressed as a linear formula

of the values of several other data fields x, the so-called predictor fields or regressors: t = $b_0 + b_1 * x_1 + ... + b_n * x_n$.

Logistic
(module: **Regressions Analysis***)*
In logistic regression, the probability of the '1'-value of a two-valued target field t is expressed as a formula of the values of several other data fields x, the so-called predictor fields or regressors. The formula has the form: proba(t=1) = $1/(1+e^{b0+b1*x1+...+bn*xn})$.

Look and feel
(module: **Workbench***)*
You can adapt the workbench design and style (look and feel) to your preferences and to your operating system. You can change between a 'MS-Windows' style, a 'Unix-Motif' style and a system independent 'Java native' ('metal') style. Do not select 'windows' if you are running on MAC OS, Unix or Linux.

Mapped Name
(module: **Statistics and Distributions***)*
Mapped field value names as they have been read from an auxiliary name mapping table.

Max. #deviations
(module: **Deviation Detection***)*
Keep the result size manageable by limiting the maximum number of deviation patterns to be detected. If more deviation patterns can be found, only the strongest ones of them are kept.

Max. number of active fields
(module: **Data Import***)*
The maximum desired number of active data fields. If the number of currently active fields exceeds this value, some of them will be deactivated. The software decides autonomously which fields are deactivated, based on the number of missing values, the number of different values and field-field correlations.

Max. number of iterations
(module: **SOM Models***)*
Limit the possible number of SOM iterations. Within one SOM iteration, the SOM training algorithm performs one scan over all training data records and uses each record for adapting the neuron weights of the best matching neuron and its neighbors.

Max. number of selected data rows
(module: **Workbench***)*
From various analysis modules of the software, the user can select a data subset, display it in tabular form in a separate screen window and export it to a flat file or database

table. In this parameter, you can specify the maximum allowed number of data rows in such data subsets. Larger subsets wil be truncated. Allowed values are 100 to 100000000

Max. pattern length
(module: **Deviation Detection***)*
The maximum length of the deviation patterns to be detected.

Max. tupel length
(module: **Statistics and Distributions***)*
Upper limit for the length of the tupels to be identified, i.e. the maximum number of items per tupel.

Maximum neighbor distance
(module: **SOM Models***)*
The maximum Euclidean distance of neighbored neurons in the SOM net over which adaptions to one neuron influence the neighbored neuron.

Maximum Number of different textual values per field
(module: **Data Import***)*
Define a maximum number N of different textual values (categories) per data field. Whenever a textual field has more than N different values, only the N most frequent of them will be kept, all other ones will be grouped into the category 'others'.

Maximum textual value length
(module: **Data Import***)*
Specify the maximum number of characters in textual values. Longer textual values will be truncated in the compressed data.

MC conf
(module: **Associations Analysis***)*
MC conf stands for 'Monte Carlo significance verification confidence'. This measure indicates how sure one can be that the given association contains a statistically significant rule within the data and is not a product of 'hazard', that means random noise in the data. The measure is calculated by trying to find associations with similar support, lift and purity values in simulated artificial data which contain the same items with the same item frequencies as the original data, but no correlations between the items.

Median
(module: **Statistics and Distributions***)*
The median of the value distribution, that means the smallest value such that 50% of the data records or groups have a value which is smaller or equal. For irreversibly binned fields, the exact median cannot be determined; instead, the mid point of the interval containing the median is returned.

Memory usage limit (MB)
(module: **Multivariate Exploration and Split Analysis***)*
Upper limit (in MB) for the RAM to be used by the automized series of split analysis tasks to be deployed.

Min. #affected records
(module: **Deviation Detection***)*
A minimum threshold for the number of data records in which a deviation pattern occurs. Deviation patterns which occur less frequently in the data will not be shown.

Min. deviation increase
(module: **Deviation Detection***)*
A minimum threshold for the increase in deviation strength when expanding patterns be adding another part (item). If this threshold is X, then only those patterns will be shown whose deviation strength is at least X times the deviation strength of each 'parent' pattern which can be obtained from the initial pattern by removing one part (item).

Min. deviation strength
(module: **Deviation Detection***)*
A minimum threshold for the strength of the deviation patterns to be detected. The strength of a deviation is the inverse of the deviation's lift value. For example, if a combination (A,B) of two data field values A and B occurs in 0.02% of all records, and if A and B alone occur in 20% respectively 10% of the data records, then the deviation strength of the pattern (A,B) is 100 since 0.02% is 100 times less than the expected occurrence frequency of 20% * 10% = 2%.

Min. tupel support
(module: **Statistics and Distributions***)*
Minimum tupel support. The support of a tupel is the number of data groups in which all items of the tupel occur.

Minimum textual value frequency
(module: **Data Import***)*
This parameter defines a lower boundary for the number of data records or data groups on which a value of a non-numeric data field must occur for being tracked as a separate field value and a separate bar in histogram charts. Less frequent values will be grouped into the category 'others'.

Minimum tupel purity
(module: **Statistics and Distributions***)*
Minimum purity of the tupels to be detected. The purity of a tupel is the tupel's occurrence frequency divided by the occurrence frequency of the tupel's most frequent item.

Model name
(modules: **Associations Analysis, Sequential Patterns, SOM Models, Regressions Analysis, Decision Trees***)*
File name under which the generated data mining model or analysis result will be stored on disk. The file name suffix determines the file format: .xml and .pmml produce a PMML model, .sql creates an SQL SELECT statement, .txt and .mdl create a flat text file.

Mouse-over help text dismiss delay
(module: **Workbench***)*
Most labels, menu items, buttons, input fields and table column headers in the graphical workbench have a 'mouse-over' function showing a context-sensitive pop-up help text. This Parameter specifies for how many seconds the help text is shown.

Mouse-over help text initial delay
(module: **Workbench***)*
Most labels, menu items, buttons, input fields and table column headers in the graphical workbench have a 'mouse-over' function showing a context-sensitive pop-up help text. This Parameter specifies how many seconds after placing the mouse pointer the help text pops up.

Mouse-over help text reshow delay
(module: **Workbench***)*
Most labels, menu items, buttons, input fields and table column headers in the graphical workbench have a 'mouse-over' function showing a context-sensitive pop-up help text. This Parameter specifies for how many seconds the help text cannot be reshown after it has been shown once.

Name mappings
(module: **Data Import***)*
A name mapping defines more readable textual values (e.g. product names) for the original values (e.g. product IDs) of a data field. A name mapping definition must contain the file or table name (optionally preceeded by the directory path or jdbc connection), the names of the fields (columns) containing the original and the mapped value, and the field name of the main data source to which the name mapping applies.

Negated items
(modules: **Associations Analysis, Sequential Patterns***)*
Negative items are items for which the complement, i.e. the fact that the item does NOT occur, should be treated as a separate item. For example, if the item 'OCCUPATION=Manager' is added to the list of negative items, then the item 'OCCUPATION!=Manager' is created, and its support is the complement of the support of 'OCCUPATION=Manager'.

No Negative Values
(module: **Time Series Analysis***)*
Restrict the allowed range for the predicted time series values to values equal or greater than zero.

Nominal value selection mode
(module: **SOM Models***)*
Method for selecting the 'best' nominal value which is shown in the SOM cards for nominal data fields.

Null-value string
(module: **Workbench***)*
If a non-empty string is specified for this parameter, then this string will be interpreted as 'n/a' ('invalid or missing value') whenever it occurs as the value of a data field.

Number of active fields
(module: **Data Import***)*
The number of currently activated data fields (not counting the entity field).

Number of items
(module: **Sequential Patterns***)*
The total number of items in the sequences to be detected. An item is one elementary peace of information, that means an atomic part within the sequential pattern

Number of patterns
(module: **Associations Analysis***)*
Keep the result size manageable by limiting the maximum number of associations to be detected. If more associations can be found, only the 'best' ones of them are kept. The criterion for selecting the 'best' associations can be defined using the radio button 'Sorting criterion'.

Number of regressors
(module: **Regressions Analysis***)*
The total number of data fields which appear on the left hand side of the regression equation which predicts target field values.

Number of sequences
(module: **Sequential Patterns***)*
Keep the result size manageable by limiting the maximum number of sequences to be detected. If more sequences can be found, only the 'best' ones of them are kept. The criterion for selecting the 'best' sequences can be defined using the radio button 'Ranking criterion'.

Number of threads
(module: **Alle***)*
Specify an upper limit for the number of parallel threads used for reading and compressing the data. If no number or a number smaller than 1 is given here, the maximum available number of CPU cores will be used in parallel.

Number of values or intervals
(module: **Data Import***)*
Determine the number of separately treated values or value ranges. Allowed values are 2...100 for numeric fields and 0...100 for textual fields.

Numeric field
(module: **Data Import***)*
A data field which is to be treated as numeric field. If it contains textual values, these values will be ignored, i.e. considered as missing values.

Numeric field weight
(module: **SOM Models***)*
Per default, each numeric data field contributes with the same weight factor (of 1) to the distance calculations between neurons and data records as the Boolean and textual fields. You can define a higher or lower weight factor for the numeric fields compared to Boolean and textual fields using this parameter. Note that weight settings for specific fields overwrite this general setting, the weight factors are not multiplied.

Numeric precision (digits)
(module: **Data Import***)*
Specify the maximum numeric precision, i.e. the maximum number of digits that will be regarded when reading numeric values. With the precision of 3, for example, the number 55555 will be stored as 55600 and -1.23456e-17 as -1.23e-17.

Only entity IDs
(module: **Sequential Patterns***)*
If 'only entity IDs' is checked, the 'Show' and 'Export' buttons will show resp. export only the entity IDs of the supported entities. If 'entire records' is checked, the 'Show' and 'Export' buttons will show resp. export the supported entities with all their available data fields.

Operator
(module: **Data Import***)*
The operator which will be applied on the existing input field(s) and/or the existing value(s) in order to create the value of the computing field.

Optimize the control data
(module: **Multivariate Exploration and Split Analysis***)*

Create a subset of the current control data set. The subset is aimed to be as representative as possible for the current test data set on all data fields which are not marked 'Target' (T) and for which the user has not manually selected different value ranges for the test and the control data.

Other values
(module: **Statistics and Distributions***)*
Total frequency of all textual values which were not counted as a separate category but summarized under 'others'.

Overall RMSE
(modules: **SOM Models, Regressions Analysis***)*
Root mean squared mapping error of the SOM net on the entire data

Parameter file
(modules: **Associations Analysis, Sequential Patterns, SOM Models, Regressions Analysis, Decision Trees***)*
File name under which the current parameter settings will be stored on disk.

Parent support ratio
(module: **Associations Analysis***)*
The acceptable support growth when comparing a given association to its parent associations. A parent association (of n-1 items) will be rejected if its support is less than the support of the current association (of n items) multiplied by the minimum parent support ratio. The effect of this filter criterion is that it reduces the number of detected associations by removing all sub-patterns of long associations whenever the sub-patterns have a support which is not strongly larger than the support of the long association.

Pattern length
(module: **Associations Analysis***)*
The length of an association is the number of items which form the association. When specifying the parameters for an associations training, you should always specify an upper boundary for the desired association lengths, otherwise the training can take extremely long time.

Perfect tupel frequency threshold
(module: **Workbench***)*
Default setting for the minimum required frequency above which a tupel of several items can be considered as a perfect tupel. Must be an integer larger than 1.

Perfect tupel purity threshold
(module: **Workbench***)*
Default setting for the minimum purity at which a tupel of several items is considered as

a perfect tupel. Must be a number between 0.5 and 1.0. For the definition of purity: see definition in module associations analysis

Perfect Tupels
(module: **Statistics and Distributions***)*
Detect (almost) perfect item tupels in the data, i.e. value combinations of textual set-valued data fields which appear (almost) always together.

Period
(module: **Time Series Analysis***)*
Presumed cycle length of the seasonal (periodic) part of the time series in units of the time step between adjacent data points.

PMML version
(module: **Workbench***)*
The software can create and export data mining models in the vendor independent PMML format (see http://www.dmg.org/pmml). This parameter defines, which version of PMML should be created.

Positions of required items
(module: **Sequential Patterns***)*
The required item type indicates at which position within a sequence the item can occur.If the type is 'Sequence start', the item must occur in the sequence's first item set.If the type is 'Sequence end', the item must occur in the sequence's last item set.If the type is 'Anywhere', the item can occur anywhere within the sequence.

Prediction error (RMSE)
(modules: **Regressions Analysis, SOM Models***)*
Root mean squared prediction error of the regression model on the training data

Primary sorting criterion
(module: **Workbench***)*
The selection box 'Primary sorting criterion' is an option that can be activated when exporting in-memory data objects into a text file on disk. When activated, the option sorts the exported data rows by ascending or descending values of the data field selected in the box.

Purity
(module: **Associations Analysis***)*
The purity of an association is the ratio between the association's support and the support of the most frequent item within the association. A purity of 1 indicates a 'perfect' group: each single item of the transaction occurs in a transaction if and only if also all the other items of the association occur in that transaction.

Purity threshold for perfect tupels
(module: **Workbench***)*
Default setting for the minimum purity at which a tupel of several items is considered as a perfect tupel. Must be a number between 0.5 and 1.0. For the definition of purity: see definition in module associations analysis

Quotation mark (default)
(module: **Workbench***)*
If this parameter in the data import settings is set to 'double quote' (or 'single quote'), then double (or single) quotes around field values are removed per default for all input data fields. If this parameter is set to 'none', then double or single quotes around field values are only removed if ALL values of the field are surrounded by the same quotes; in addition, numeric values surrounded by quotes are interpreted as textual values in this case.

Read data
(module: **Data Import***)*
This button starts reading the original data source and transforming the data into a compressed binary data object which resides in memory.

Records for guessing field types
(module: **Data Import***)*
When reading input data from flat files or spreadsheets, the data source does not provide meta data information on the types of data (integer, Boolean, floating point, textual) to be expected in the available data columns. Therefore, a presumable data type has to be derived from looking at the data fields actual content. The parameter 'Number of records for guessing field types' determines, how many leading data rows are read from the data source for guessing data field types.

Refresh
(module: **Statistics and Distributions***)*
Refresh the screen, for example in order to adapt to a changed screen size.

Regression coefficient
(module: **Regressions Analysis***)*
Regression coefficients are the weight prefactors with which the different regressors enter into the regression equation.

Regression method
(module: **Regressions Analysis***)*
The software supports two regression methods: linear regression and logistic regression. In linear regression, the value of a numeric target field t is expressed as a linear formula of the values of several other data fields x, the so-called predictor fields or regressors: $t = b_0 + b_1{*}x_1 + ... + b_n{*}x_n$.In logistic regression, the probability of the '1'-value of a two-valued

target field t is expressed as a formula of the kind: proba(t=1) = $1/(1+e^{b0+b1*x1+...+bn*xn})$.

Regression Model
(modules: **Workbench, Data Import, Regressions Analysis***)*
In this panel, you can visualize and introspect the results of a regression training run, that means the regression coefficients and model quality measures such as RMSE or R-squared values.

Regression Scoring
(modules: **Workbench, Data Import, Regressions Analysis***)*
In this module, you specify the parameters and settings which are to be used for applying a regression model to new data.

Regression Training
(modules: **Workbench, Data Import, Regressions Analysis***)*
A regression training establishes a formula which predicts the value of one single data field from the values of some other fields within the training data. In the regression training panel, you specify the parameters and settings which are to be used for the next regression training run. Furthermore, you can store your parameter settings, manage them in a repository and later retrieve and reuse them. In the lower part of the panel, you can start and stop a regression training run and monitor its progress and its predicted run time.

Regressor
(module: **Regressions Analysis***)*
A regressor is a data field which appears on the left hand side of the regression equation and whose values serve to predict the target field value.

Regressor fields
(module: **Regressions Analysis***)*
Upper limit for the number of regressors which can enter into the regression model.

Rel. difference
(modules: **Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
relative difference: |#selected - #expected| / #expected.

Rel.diff
(module: **SOM Models***)*
Maximum relative difference to the field's overall value distribution: the SOM card shows the nominal value for which the ratio between its actual frequency within the records mapped to the given neuron and its expected frequency is maximum.

Relative difference
(modules: **Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
relative difference: |#selected - #expected| / #expected.

Relative difference
(module: **SOM Models***)*
Maximum relative difference to the field's overall value distribution: the SOM card shows the nominal value for which the ratio between its actual frequency within the records mapped to the given neuron and its expected frequency is maximum.

Relative Frequency
(module: **Statistics and Distributions***)*
Fraction of all data records or data groups which contain the value

Relative item support
(modules: **Associations Analysis, Sequential Patterns***)*
The relative support of an item is the item's absolute support divided by the total number of transaction (groups). In other words, the relative support is the a-priori probability that the item occurs in a randomly selected transaction.

Relative support
(module: **Associations Analysis***)*
The relative support of an association is the absolute support divided by the total number of groups (transactions), that means the a-priori probability that an arbitrary group supports the association. When specifying the parameters for an associations training, you should always specify an lower boundary for the absolute or relative support, otherwise the training can take extremely long time.

Relative support
(module: **Sequential Patterns***)*
The relative support of the sequence, that means the fraction of all entities (transaction groups) in which the sequence occurs

Reporting Preferences
(module: **Workbench***)*
Preference settings for the visual report designer and for creating HTML and PDF reports.

Required items
(modules: **Deviation Detection, Associations Analysis, Sequential Patterns***)*
Required items are items which must occur in each detected pattern. If several item patterns are specified within one 'required group', at least one of them must appear in each detected deviation, association or sequence. In the Associations and Sequences

training modules, up to 3 different groups of required items can be specified. In this case, the detected patterns will contain at least one item out of every specified group. Each item specification can contain wildcards (*) at the beginning, in the middle and/or at the end.

Required items - permitted position
(module: **Sequential Patterns***)*
The required item type indicates at which position within a sequence the item can occur.If the type is 'Sequence start', the item must occur in the sequence's first item set.If the type is 'Sequence end', the item must occur in the sequence's last item set.If the type is 'Anywhere', the item can occur anywhere within the sequence.

Result file
(modules: **Associations Analysis, Sequential Patterns, SOM Models, Regressions Analysis, Decision Trees***)*
File name under which the generated data mining model or analysis result will be stored on disk. The file name suffix determines the file format: .xml and .pmml produce a PMML model, .sql creates an SQL SELECT statement, .txt and .mdl create a flat text file.

RMSE
(modules: **Regressions Analysis, SOM Models***)*
Root mean squared prediction error of the regression model on the training data

Row filter criterion
(module: **Data Import***)*
A sampling criterion or SQL WHERE clause. For example, the criterion '10%' creates a random sample of about 10% of all data rows. The criterion '!10%' creates the complementary subset containing all records which the criterion '10%' would have blocked. The criterion WHERE GENDER='M' selects all data rows whose 'GENDER' value is 'M'.

R^2
(module: **Regressions Analysis***)*
R^2 is a measure for the predictive power of the regression model. R^2 near 1 means that the model is able to predict the target values almost perfectly, R^2 near 0 means that the model is almost useless.

Screen height
(module: **Workbench***)*
Default height of the main workbench window (in pixels). Allowed values are 480 to 1500

Screen width
(module: **Workbench***)*

Default width of the main workbench window (in pixels). Allowed values are 640 to 2000

Secondary sorting criterion
(module: **Workbench***)*
The selection box 'Secondary sorting criterion' defines an additional sorting criterion which applies for sorting data rows with identical values in the primary sorting criterion.

Selected data rows
(module: **Workbench***)*
From various analysis modules of the software, the user can select a data subset, display it in tabular form in a separate screen window and export it to a flat file or database table. In this parameter, you can specify the maximum allowed number of data rows in such data subsets. Larger subsets wil be truncated. Allowed values are 100 to 100000000

Selected records
(module: **SOM Models***)*
The number of data records mapped to the currently selected neurons.

Selected RMSE
(module: **SOM Models***)*
Root mean squared mapping error of the SOM net on the data records mapped to the currently selected neurons.

Sequence length
(module: **Sequential Patterns***)*
The desired sequence lengths of the sequences to be detected. The sequence length is the number of parts (events) separated by time steps.

Sequences Detection
(modules: **Workbench, Data Import, Sequential Patterns***)*
In this panel, you specify the parameters and settings which are to be used for the next Sequential Patterns training run. Furthermore, you can store your parameter settings, manage them in a repository and later retrieve and reuse them. In the lower part of the panel, you can start and stop a Sequential Patterns training run and monitor its progress and its predicted run time.Sequential Patterns Analysis is only possible on data on which an 'Entity' field, a 'Group' field and an 'Order' field has been defined on the 'Active fields' dialog. The Group field and the Order field can be identical; in this case, specify the field as "Order and Group' field.

Sequences Model
(modules: **Workbench, Data Import, Sequential Patterns***)*

A sequences model is a collection of sequential patterns which have been detected during a sequences training run on a training data set. The model can be applied to a new data source in a sequences scoring step. In the sequences model panel, you can visualize and introspect the results of a Sequential Patterns training run. You can display the results in tabular form, sort, filter and export the filtered results to flat files or into a table in a RDBMS. Furthermore, you can calculate additional statistics for the support of selected sequential patterns.

Sequences Scoring
(modules: **Workbench, Data Import, Sequential Patterns***)*
A Sequences Scoring presents new data records to a previously trained Sequential Patterns model. A Sequential Patterns model is a collection of sequences of events which were observed in the data on which the model was trained. The scoring relates sequences from the model with data records from the new data. This can be done in two ways. The first way examines one or more selected data records (e.g. all purchases of one single customer) and returns all sequences which are partially or fully supported by the selected records. The second way examines one or more selected sequences and returns all records (e.g. all customers) that partially or fully support the selected sequences. You can store and retrieve both the parameter settings for Sequences Scoring and the scoring results in the form of XML or flat text files.

Set frequencies
(module: **Sequential Patterns***)*
The absolute supports of the item sets which form the sequence. (the first number corresponds to set1, the second to set2, etc.) A star (*) after the number indicates that the set belongs to the core of the sequence. The core of a sequence is the smallest possible subsequence of item sets of the sequence which has the same support as the entire sequence.

Significance
(modules: **Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*

Skewness
(module: **Statistics and Distributions***)*
The sample skewness of the value distribution. Note: the sample skewness slightly differs from population skewness (e.g. MS Excel's 'Skewness').

Smoothing
(module: **Time Series Analysis***)*
Number of time points used for calculating the moving average trend line.

SOM cards per row
(module: **SOM Models***)*
The number of SOM cards placed in one row. Reduce this number for obtaining larger graphs.

SOM Model
(modules: **Workbench, Data Import, SOM Models***)*
A SOM model is a neural network which has been trained in a preceeding SOM training run on some training data and which has 'learned' the training data during that training. You can visualize and introspect the SOM model with its SOM cards. You can explore different regions of the SOM map, explore the statistics of these regions and export data records mapped to these regions to flat files or into a table in a RDBMS. The model can be applied to a new data source in a SOM scoring step, for example in order to predict one or more data fields' values which are unknown in the new data.

SOM Scoring
(modules: **Workbench, Data Import, SOM Models***)*
A SOM Scoring presents new data records to a previously trained Self Organizing Map (SOM) model. A SOM model is a neural network which represents the data by means of a square grid of neurons. The scoring can be used to predict missing values in the new data, to classify the new data records as deviations, or to assign them to clusters (segments). You can store and retrieve both the parameter settings for a SOM scoring and the scoring results in the form of XML or flat text files.

SOM Training
(modules: **Workbench, Data Import, SOM Models***)*
A SOM training task specifies the parameters and settings which are to be used for the next SOM training run. In the SOM Training Task panel, you can store your parameter settings, manage them in a repository and later retrieve and reuse them. In the lower part of the panel, you can start and stop a SOM training run and monitor its progress and its predicted run time.

Sorting criterion
(modules: **Associations Analysis, Sequential Patterns***)*
The ranking criterion which is used to sort out certain detected patterns (associations or sequences) when the total number of detected patterns becomes larger than the user-defined maximum desired number. Possible values are Support, Lift, Purity, Core item purity, Weight or Trend. Weight is only allowed if a weight field has been defined on the input data. Trend is only allowed if an order field has been defined on the input data.

Split Analysis
(modules: **Workbench, Data Import, Multivariate Exploration and Split Analysis***)*
Split Analysis is data analysis approach in which two data subsets are selected: a 'test'

data set and a 'control' data set. In many use cases, the test data set comprises a data subset which have a certain property in common, for example all men, all customers below the age of 30, all vehicles produced after an improvement measure has been effectuated, etc. The first goal of the analysis is to select a suitable control group which is representative for the test group in all attributes except the ones used for defining the test group. The second goal is to find and quantify significant differences between the test data subset and the control data subset.

Standard codepage
(module: **Workbench***)*
Whenever a data source contains non-standard-English characters (such as î, ä, é, € etc.) you must specify in which encoding scheme (codepage) the data have been encoded, otherwise these characters will not be displayed correctly. If you do not know the encoding scheme, you have to try out various choices.

Standard deviation of relative difference
(modules: **Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
Standard deviation of relative difference. This value indicates how exactly the relative difference can be calculated.

Std. deviation
(module: **Statistics and Distributions***)*
The sample standard deviation of the value distribution (i.e. the 'n' and not the 'n-1' standard deviation!)

Std.dev.(rel.diff.)
(modules: **Multivariate Exploration and Split Analysis, Multivariate Exploration and Split Analysis***)*
Standard deviation of relative difference. This value indicates how exactly the relative difference can be calculated.

Store the load task as XML file
(module: **Data Import***)*
If this check box is marked, the current data load settings are written into a persistent XML file. The settings in this XML file can later be applied to any new data source of the same structure as the original data source.

Summary result file
(module: **Multivariate Exploration and Split Analysis***)*
File name of a TAB-separated tabular text file in which the summary result of the series of split analysis tasks will be written. The file will contain one row per single split analysis. If no value is given here, no summary result file will be created.

Superset
(module: **Associations Analysis***)*
If 'superset' is checked, the 'Show', 'Explore' and 'Export' buttons will handle each data record or group which supports at least one of the selected associations. If 'intersection' is checked, the 'Show', 'Explore' and 'Export' buttons will only handle those data groups which support *all* selected associations.

Superset
(module: **Sequential Patterns***)*
If 'superset' is checked, the 'Show', 'Explore' and 'Export' buttons will cover each entity which supports at least one of the selected sequences. If 'intersection' is checked, the 'Show', 'Explore' and 'Export' buttons will only cover those entities which support *all* selected sequences.

Suppressed field
(module: **Data Import***)*
A data field which will be completely ignored.

Suppressed items
(modules: **Deviation Detection, Associations Analysis, Sequential Patterns***)*
Suppressed items are items which are completely ignored during the patterns analysis and which should never occur in the detected patterns. Each item specification can contain wildcards (*) at the beginning, in the middle and/or at the end.

Target (not to be optimized)
(module: **Multivariate Exploration and Split Analysis***)*
Target fields are those visible fields whose field value differences between test and control data will be ignored during the control data optimization. These fields are the 'target' fields of the hypothesis test. The aim of the test is to find out whether there are significant value distribution differences between the test and control data on these fields.

Target field
(modules: **SOM Models, Reporting***)*
Specify the name of the target field if you want to use the SOM method for predicting the values of one single data field.

Target field
(modules: **Regressions Analysis, Decision Trees***)*
The name of the target field, that means the name of the field whose values are to be predicted from the values of the other data fields.

Target field weight
(module: **SOM Models***)*
Per default, each data field contributes with the same weight factor (of 1) to the distance

calculations between neurons and data records. You can assign a higher weight factor to the target field.

Taxonomies (hierarchies)
(module: **Data Import***)*
A taxonomy is the definition of a category hierarchy. For example, such a hierarchy could define the two products 'butter' and 'cheese' as members of the category 'milk products', and 'milk products' as a sub-category of 'food'. Taxonomy definitions can be read from flat files or database tables. A taxonomy definition must contain the file or table name (optionally preceeded by the directory path or jdbc connection), the names of the fields (columns) containing the parent and the child categories, and the field name of the main data source to which the taxonomy applies.

Temporary file directory
(module: **Workbench***)*
In this directory, temporary dump files will be stored. Dump files are created when reading data from very large data sources.

Test data
(module: **Multivariate Exploration and Split Analysis***)*
The currently selected test data subset in a test-control data analysis. The goal of the analysis is to detect and quantify systematic deviations in the field value distribution properties between the test data subset and the control data subset

Textual field
(module: **Data Import***)*
A data field whose values are to be treated as textual (categorical) values even if they are numeric values.

Textual resource file
(module: **Workbench***)*
File in which all textual resources needed by the workbench are stored: labels of menus, input fields and buttons, context sensitive help texts, glossary entries etc. If you want to customize the software, you can work with personalized versions of the default file IA_texts.xml.

Time Series Analysis and Forecast
(modules: **Workbench, Data Import, Time Series Analysis***)*
In the Time Series panel, time series can be explored and forecasts can be calculated using various forecasting algorithms.This module can only be started on data which fulfill the following requirements:i) An order field has been defined in the 'Active fields' dialog. This field will be the x-axis field in the time series charts.ii) A weight/price field has been defined in the 'Active fields' dialog. This field will be the y-axis field in the time series

charts.iii) Not more than two further active fields exist (plus optionally a group field). All other fields have been deactivated in the 'Active fields' dialog.

Time step limits
(module: **Sequential Patterns***)*
Time step limits define which time step size is permissible between adjacent parts (item sets) of a sequence.

Time/order field
(module: **Data Import***)*
A data field should be marked as 'time/order field' if it does not contain an property of the entity to be analyzed but the time stamp or step identifier at which the entity's properties in the other data fields of the current data row have been recorded. For some data mining functions, the specification of a time/order field is required (e.g. sequence analysis, time series prediction), other data mining functions will ignore any time/order information (e.g. associations analysis).

Tooltip dismiss delay
(module: **Workbench***)*
Most labels, menu items, buttons, input fields and table column headers in the graphical workbench have a 'mouse-over' function showing a context-sensitive pop-up help text. This Parameter specifies for how many seconds the help text is shown.

Tooltip initial delay
(module: **Workbench***)*
Most labels, menu items, buttons, input fields and table column headers in the graphical workbench have a 'mouse-over' function showing a context-sensitive pop-up help text. This Parameter specifies how many seconds after placing the mouse pointer the help text pops up.

Tooltip reshow delay
(module: **Workbench***)*
Most labels, menu items, buttons, input fields and table column headers in the graphical workbench have a 'mouse-over' function showing a context-sensitive pop-up help text. This Parameter specifies for how many seconds the help text cannot be reshown after it has been shown once.

Total time window
(module: **Sequential Patterns***)*
The desired time gap between the first and the last part (event) of the sequences to be detected.

Trace file
(module: **Workbench***)*

Name of the trace file to which the software writes success, progress, warning and error messages. Choose a qualified file name such as 'C:\IA\IA_trace.log', or the string 'stdOut' if you want to trace to the black console window.

Trace level
(module: **Workbench***)*
The frequency (intensity) of protocol output. The higher, the more protocol output is produced. Allowed levels are 0 to 4. In level 0, no protocol output is produced. In level 4, the protocol output might become very large if you are working on large data.

Tracked items
(module: **Associations Analysis***)*
Tracked items are items whose occurrence rate is tracked and shown for every detected association. The tracked rate indicates the probability that the tracked item occurs in a data record or group which supports the current association.

Training data
(modules: **Associations Analysis, Sequential Patterns, SOM Models, Regressions Analysis, Decision Trees***)*
Training data are a data collection on which a data mining model is being trained. During the training, the model 'learns' certain rules, interrelations and dependencies between the differen data fields of the training data. After the training, the model can be applied to new data, for example in order to predict missing field values or in order to classify or cluster new data reords. This is called 'scoring'.

Tree Preferences
(module: **Workbench***)*
Preference settings for Decision and Regression Tree (model training and application)

Tree Training
(modules: **Workbench, Data Import, Decision Trees***)*
A decision tree training establishes a hierarchical, tree-like set of Boolean predicates which describe the typical behavior of one single 'target' attribute in the training data. In the tree training panel, you specify the parameters and settings which are to be used for the next decision tree training run. Furthermore, you can store your parameter settings, manage them in a repository and later retrieve and reuse them. In the lower part of the panel, you can start and stop a decision tree training run and monitor its progress and its predicted run time.

Trend damping
(module: **Time Series Analysis***)*
Damping factor applied when projecting current trend into the future. If, for example, the trend damping factor is 0.9, if the time series data are recorded monthly, if the current trend is a seasonally corrected month-to-month increase dx and if the current month's

seasonally corrected value is x, then the seasonally corrected projected values for the next 3 months will be x+0.9*dx, x+(0.9+0.81)*dx, x+(0.9+0.81+0.729)*dx.

Undo
(module: **Multivariate Exploration and Split Analysis***)*
Undo the previous control data optimization. That means, reactivate all available control data records.

Values
(module: **Data Import***)*
Define a maximum number N of different textual values (categories) per data field. Whenever a textual field has more than N different values, only the N most frequent of them will be kept, all other ones will be grouped into the category 'others'.

Variant elimination
(module: **Data Import***)*
A variant elimination replaces several spelling variants or misspellings, several case variants and/or several synonyms for identical things or concepts by one single 'canonical' form. Variant eliminations can be specified for all textual data fields. Variants can be defined either by listing the variants one by one or by using regular expressions (pattern matching).

Verif. confidence
(module: **Associations Analysis***)*
Verification runs serve to assess whether the detected association or sequential patterns are statistically significant patterns or just random fluctuations (white noise). For each verification run, a separate data base is used. Each data base is generated from the original data by randomly assigning each data field's values to another data row index within the same data field. This approach is called a permutation test. The effect is that correlations and interrelations between different data fields are completely removed from the data. If one finds association or sequential patterns on a permuted data base, one can be sure that one has detected nothing but noise. One can record and trace the measure triples (pattern length, support, lift) of all detected noise patterns. The edge of the resulting point cloud defines the intrinsic 'noise level' of the original data. Patterns detected on the original data can only be considered significant if their corresponding measure triples are well above the noise level. These patterns have a verification confidence close to 1.

Verification confidence (of an association pattern)
(module: **Associations Analysis***)*
Verification runs serve to assess whether the detected association or sequential patterns are statistically significant patterns or just random fluctuations (white noise). For each verification run, a separate data base is used. Each data base is generated from the original data by randomly assigning each data field's values to another data row index within the same data field. This approach is called a permutation test. The effect is that correlations

and interrelations between different data fields are completely removed from the data. If one finds association or sequential patterns on a permuted data base, one can be sure that one has detected nothing but noise. One can record and trace the measure triples (pattern length, support, lift) of all detected noise patterns. The edge of the resulting point cloud defines the intrinsic 'noise level' of the original data. Patterns detected on the original data can only be considered significant if their corresponding measure triples are well above the noise level. These patterns have a verification confidence close to 1.

Verification run
(modules: **SOM Models, Decision Trees***)*
In addition to the main training run, you can start 0 to 9 verification runs. Each verification run is a separate training run with the same parameters as the main training run but a different seed value for the random number generator. The purpose of verification runs is to generate stability and reliability information for the model created by the main training run.

Verification run
(modules: **Associations Analysis, Sequential Patterns***)*
Verification runs serve to assess whether the detected association or sequential patterns are statistically significant patterns or just random fluctuations (white noise). For each verification run, a separate data base is used. Each data base is generated from the original data by randomly assigning each data field's values to another data row index within the same data field. This approach is called a permutation test. The effect is that correlations and interrelations between different data fields are completely removed from the data. If one finds association or sequential patterns on a permuted data base, one can be sure that one has detected nothing but noise. One can record and trace the measure triples (pattern length, support, lift) of all detected noise patterns. The edge of the resulting point cloud defines the intrinsic 'noise level' of the original data. Patterns detected on the original data can only be considered significant if their corresponding measure triples are well above the noise level.

Verification runs
(modules: **SOM Models, Decision Trees***)*
In addition to the main training run, you can start 0 to 9 verification runs. Each verification run is a separate training run with the same parameters as the main training run but a different seed value for the random number generator. The purpose of verification runs is to generate stability and reliability information for the model created by the main training run.

Verification runs
(modules: **Associations Analysis, Sequential Patterns***)*
Verification runs serve to assess whether the detected association or sequential patterns are statistically significant patterns or just random fluctuations (white noise). For each verification run, a separate data base is used. Each data base is generated from the original data by randomly assigning each data field's values to another data row index

within the same data field. This approach is called a permutation test. The effect is that correlations and interrelations between different data fields are completely removed from the data. If one finds association or sequential patterns on a permuted data base, one can be sure that one has detected nothing but noise. One can record and trace the measure triples (pattern length, support, lift) of all detected noise patterns. The edge of the resulting point cloud defines the intrinsic 'noise level' of the original data. Patterns detected on the original data can only be considered significant if their corresponding measure triples are well above the noise level.

Visible SOM cards
(module: **SOM Models***)*
Select the data fields for which you want to see SOM cards in the main panel above. Per default, the SOM cards for the 20 data fields with highest field importance numbers are shown.

Web browser call command
(module: **Workbench***)*
For accessing online help, the software must start an external web browser. This parameter contains the calling command for this browser. There are default settings for several operating systems. Therefore, you should only modify this parameter if you are unable to use the online help with the default settings.

Weight
(module: **Associations Analysis***)*
The weight of an association is the mean weight of all data records (or data groups) which support the association. The weight of a data group is either the sum, the average, the minimum, or the maximum of the weight field values, or the number of records, of all input data records which form the group. The actual computation variant depends on the aggregation mode that has be set for the weight field in the input data panel (sum,mean,max,min, or count).

Weight/price field
(module: **Data Import***)*
A data field should be marked as 'weight/price field' if it contains the price, cost, weight, or another numeric quantity which characterizes the 'importance' of the properties given in the other data fields of the current data row.

Width of the neural net
(modules: **SOM Models, Reporting***)*
The number of neurons in direction x. Should be a number between 4 and 100